Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Penetrační testy v IP telefonii

Penetration Tests in IP Telephony

2015 **Bc. Jozef Záhon**

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání diplomové práce

Student: **Bc. Jozef Záhon**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2601T013 Telekomunikační technika

Téma: Penetrační testy v IP telefonii
Penetration Tests in IP Telephony

Zásady pro vypracování:

Metoda penetračních testů se v současnosti využívá nejčastěji pro detekci úrovně zabezpečení síťových prvků v dané infrastruktuře. V komerční i nekomerční oblasti existuje několik nástrojů, které tyto testy realizují. Cílem práce je jednak analyzovat současný stav těchto nástrojů, převážně z pohledu podpory penetračních testů pro SIP prvky IP telefonie. Dále definovat a klasifikovat jaké techniky či aplikace by byly pro vytvoření SIP VoIP penetračních testů nejvhodnější a vytvořit zásady zabezpečení vycházející z výsledků testů. Obsahem práce bude i praktická ukázka vybraných aplikací, vhodných pro realizaci SIP penetračních testů v IP telefonii.

1. Popis problematiky penetračních testů v síťových službách.
2. Analýza nástrojů pro generování penetračních testů: Nessus, OpenVAS.
3. Klasifikace vhodných penetračních testů pro SIP prvky IP telefonie - Skenování a monitoring, DoS útok, krádež registrace, Man-in-the-Middle, SPIT.
4. Realizace testů pomocí zvolených aplikací.
5. Definování zásad zabezpečení na základě výsledků praktického testování.

Seznam doporučené odborné literatury:

1. Mark Collier, David Endler - Hacking Exposed Unified Communications & VoIP Security Secrets & Solutions, Second Edition, 2013, ISBN-13: 978-0071798761

2. Himanshu Kumar - Learning Nessus for Penetration Testing,

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Filip Řezáč**

Datum zadání: 01.09.2014
Datum odevzdání: 07.05.2015

doc. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prehlásenie študenta

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

V Ostrave, dňa: *5. mája 2015*

Žalm
.....
podpis študenta

Poďakovanie

Rád by som poďakoval Ing. Filipovi Řezáčovi za odbornú pomoc a konzultáciu pri vytváraní tejto diplomovej práce. Ďalej by som chcel poďakovať svojej priateľke za gramatickú korektúru a psychickú podporu.

Abstrakt

Táto diplomová práca sa zaoberá penetračnými testami so zameraním na signalizačný protokol SIP a prvky SIP infraštruktúry. Na začiatku práce je detailne popísaný protokol SIP, protokoly UDP, SDP, RTP a jeho dve zabezpečené verzie SRTP a ZRTP, využívané k prenosu multimediálneho obsahu. Nasledujúce kapitoly obsahujú popis najčastejších útokov v IP telefónii, topológie testovanej siete a nástrojov, ktoré boli zvolené k penetračným testom. V predposlednej kapitole sú realizované útoky jednotlivými nástrojmi, rozdelené do kategórií útokov z druhej kapitoly. Záverečná časť práce je venovaná bezpečnostným zásadám, vypracovaným na základe výsledkov realizovaných testov.

Kľúčové slová

Asterisk, DoS, Ettercap, Inviteflood, Kali Linux, krádež registrácie, Nessus, Nmap, MITM, OpenVAS, penetračný test, SIP, SIPp, SIPVicious, skenovanie siete, SPIT, RTP, VoIP, Wireshark

Abstract

The diploma thesis deals with penetration tests with a focus on a signalling protocol SIP and elements of SIP infrastructure. The work begins with the detailed description of the protocol SIP, protocols UDP, SDP, RTP and its two protected versions SRTP and ZRTP, which are used for the transfer of the multimedia content. Next chapters contain a description of the most frequent attacks in IP telephony, the topology of the tested network and tools that were chosen for penetration tests. In the next-to-last chapter are performed the attacks by different tools, and divided into categories of attacks from the second chapter. The last part of the thesis focuses on the security rules worked out on the basis of the tests' results.

Key words

Asterisk, DoS, Ettercap, Inviteflood, Kali Linux, registration hijacking, Nessus, Nmap, MITM, OpenVAS, penetrating test, SIP, SIPp, SIPVicious, network scanning, SPIT, RTP, VoIP, Wireshark

Zoznam použitých skratiek

Skratka	Význam
3DES	Triple Data Encryption Standard
AES	Advanced Encryption Standard
AES-CTR	Advanced Encryption Standard Counter Mode
AES-f8	Advanced Encryption Standard f8-mode
AH	Authentication Header
ACL	Access Control List
ARP	Address Resolution Protocol
B2BUA	Back-to-back User Agent
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CLI	Command Line Interface
DES	Data Encryption Standard
DH	Diffie-Hellman
DHCP	Dynamic Host Configuration Protocol
DiffServ	Differentiated Services
DNS	Domain Name System
DDoS	Distributed Denial of Service
DoS	Denial of Service
DSCP	Differentiated Services Code Point
ESP	Encapsulating Security Payload
GPL	General Public License
HTTP	Hypertext Transfer Protocol
IAX2	InterAsterisk eXchange Protocol v2
ICMP	Internet Control Message Protocol
IP	Internet Protocol
IPsec	Internet Protocol Security
KDE	K Desktop Environment
LTS	Long Term Support
LXDE	Lightweight X11 Desktop Environment
MAC	Media Access Control
MD5	Message-Digest algorithm 5
MGCP	Media Gateway Control Protocol
MITM	Man in the middle
NIPS	Network-based Intrusion Prevention System

Nmap	Network Mapper
NVT	Network Vulnerability Test
OpenVAS	Open Vulnerability Assessment System
PPTP	Point-to-Point Tunneling Protocol
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RFC	Request for Comments
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
SAS	Short Authentication String
SBC	Session Border Controller
SCCP	Skinny Client Control Protocol
SDP	Session Description Protocol
SHA	Secure Hash Algorithm
SIP	Session Initiation Protocol
SNMP	Simple Network Management Protocol
SPIT	Spam over Internet Telephony
SRTP	Secure Real-time Transport Protocol
SSH	Secure Shell
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TLS	Transport Layer Security
UA	User Agent
UC	Unified Communications
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
VLAN	Virtual Local Area Network
VoIP	Voice over Internet Protocol
ZRTP	Zimmermann Real-time Transport Protocol

Obsah

Úvod

Penetračný test je užitočným nástrojom, preverujúcim zabezpečenie siete. Audítor, vykonávajúci penetračný test, využíva techniky veľmi podobné technikám reálnych útočníkov, čím prakticky preveruje bezpečnostné mechanizmy.

Počiatočným krokom pri vonkajších penetračných testoch je obvykle zhromažďovanie informácii o testovanom subjekte. Nasleduje skenovanie adresného rozsahu, s cieľom zistiť využité adresy a na nich bežiace služby. Dôkladnejšou analýzou služieb sa stanovia medzery v zabezpečení a aplikujú sa rôzne utility slúžiace napr. ku lámaniu hesiel a podobne.

Vnútorné penetračné testy začínajú definovaním možností prístupu. Prvá možnosť využíva voľne prístupné bezdrôtové siete, ktoré poskytujú nielen anonymitu, ale aj prístup ku všetkým zariadeniam v sieti. V druhej možnosti má audítor prístup do siete organizácie, avšak bez prístupového účtu. Posledná možnosť simuluje útok na spoločnosť nelojálnym zamestnancom, kde účelom penetračných testov je odhaliť nedostatky v používaných službách, možnosti prístupu k citlivým dátam atď.

Pod pojmom sociálne inžinierstvo, je možné predstaviť si útočníka, ktorý kontaktuje zamestnancov a snaží sa tak získať citlivé informácie, akými môžu byť napr. používateľské mená a heslá k informačnému systému, prípadne ich presvedčiť k spusteniu zamaskovaného škodlivého softwaru.

Diplomová práca má za úlohu analyzovať súčasný stav nástrojov, určených k penetračným testom, predovšetkým z pohľadu podpory penetračných testov pre SIP prvky IP telefónie, realizovať testy pomocou zvolených aplikácii a na základe výsledkov testov definovať zásady zabezpečenia. Prvá kapitola je zameraná na protokol SIP, jeho vlastnosti, možnosti zabezpečenia. Čitateľ sa môže dozvedieť aj o najčastejšie používaných protokoloch v spojení s protokolom SIP, protokole UDP, ako nosnom protokole, a protokole RTP, slúžiacom k prenosu multimédií. Druhá kapitola zoznamuje čitateľa s rôznymi druhmi útokov, ktorými je možné sa v IP telefónii stretnúť. Tretia kapitola obsahuje základný popis aplikácii, použitých pri vytváraní tejto diplomovej práce. Štvrtá kapitola je pre túto prácu kľúčovou, pretože sú v nej realizované penetračné testy pomocou všetkých zvolených aplikácii. Jednotlivé testy sú rozdelené do kategórii útokov, definovaných v druhej kapitole. V záverečnej kapitole je možné dočítať sa o možnostiach zabezpečenia siete proti útokom podobným tým, ktoré boli v práci realizované.

1 Protokoly

1.1 UDP

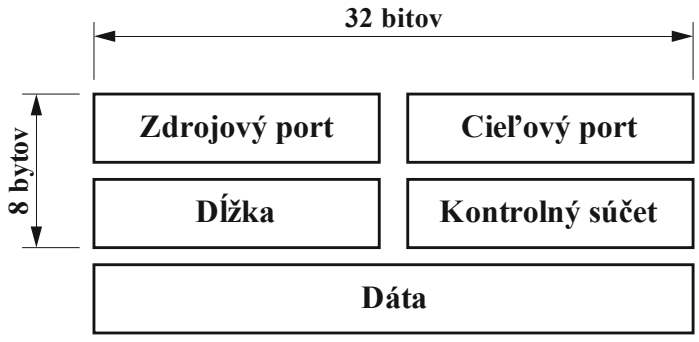
User Datagram Protocol, definovaný v RFC 768, zaobstaráva tak málo, ako protokol transportnej vrstvy musí. Okrem multiplexingu a demultiplexingu, nepridáva k IP protokolu takmer nič. Z toho dôvodu, ak sa vývojár aplikácie rozhodne použiť namiesto TCP protokol UDP, aplikácia komunikuje prakticky priamo len s IP protokolom. UDP prevezme správu z aplikačnej vrstvy, pre potreby multiplexingu a demultiplexingu vyplní zdrojový port, cieľový port a dve ďalšie polia a celý segment posunie sieťovej vrstve. Sieťová vrstva zabalí segment do IP datagramu a metódou najjednoduchšieho doručenia odošle datagram príjemcovi. V prípade úspešného doručenia použije UDP čísla portov, zdrojovú a cieľovú IP adresu k predaniu segmentu správnemu aplikačnému procesu. UDP nemá žiadny mechanizmus, kontrolujúci doručenie odoslaného datagramu. Z toho dôvodu je označený ako nespojový protokol. [20]

V porovnaní s TCP má UDP niekoľko výhod:

- bez nutnosti vytvoriť pred samotným prenosom spojenie, je odosielanie dát prostredníctvom UDP oveľa rýchlejšie,
- hlavička TCP protokolu má veľkosť 20 bytov, hlavička UDP 8 bytov,
- rýchlosť odosielania dát je limitovaná len možnosťami aplikácie, výkonom zdroja dát a prenosovou schopnosťou siete, navyše väčšina aplikácii dokáže tolerovať občasné straty v prenose.

Hlavička UDP datagramu má veľkosť 8 bytov a je zložená zo 4 častí (obr. 1.1):

- Zdrojový port, voliteľné pole označujúce port odosielajúceho procesu, na ktorý môže byť v prípade potreby odoslaná odpoveď.
- Cieľový port cieľového zariadenia.
- Dĺžka reprezentuje dĺžku UDP datagramu vyjadrenú v bytoch vrátane hlavičky.
- Kontrolný súčet počítaný, od pseudohlavičky IP paketu, cez UDP hlavičku až po koniec dát, doplnený v prípade potreby o nuly na konci tak, aby výsledná veľkosť bola presne 2 byty.



Obrázok 1.1: *Hlavička UDP datagramu*

1.2 SIP

Session Internal Protocol je textovo orientovaný protokol, definovaný v RFC 3261. Používa sa pri zostavení, udržiavaní a ukončovaní spojenia s jedným, prípadne viacerými účastníkmi. Pri jeho návrhu sa kládol dôraz najmä na jednoduchú implementáciu, rozšíriteľnosť a flexibilitu. Na rozdiel od klasického PSTN riešenia, je v SIPe použitý end-to-end koncept, čo znamená, že logika nie je uložená v sieti, ale v koncových zariadeniach. Vďaka tomu je SIP odolnejší proti chybám, má rovnakú funkcionalitu ako PSTN a sieťam umožňuje dosiahnuť vyšší výkon. Spolu so SIPom sú často používané protokoly RTP a SDP. SDP sa používa pri vyjednávaní parametrov všetkých zúčastnených zariadení a RTP na prenos hlasu, prípadne videa v paketoch. [15]

1.2.1 Typy správ v SIPe

Každá správa je tvorená hlavičkou a vlastným obsahom a zvyčajne prenášaná v samostatnom UDP datagrame. Za účelom komunikácie sa v SIPe používajú dva typy správ:

- žiadosť (metóda),
- odpoveď.

V SIPe sú definované metódy [2]:

Správa	Význam
INVITE	žiadosť odosielaná pri vytváraní nového spojenia alebo pri zmene parametrov už existujúceho spojenia.
ACK	je potvrdením prijatia odpovede na žiadosť INVITE.
BYE	touto správou sa ukončuje spojenie.
CANCEL	slúži k zrušeniu zostavovaného spojenia volajúcim v prípade, že volaný ešte nepotvrdil žiadosť INVITE.
REGISTER	obsahuje aktuálnu IP adresu a port, musí byť periodicky obnovovaná.
OPTIONS	je žiadosť o odoslanie parametrov serveru, prípadne UA.
REFER	presmeruje hovor a kontaktuje externé zdroje.
PRACK	je dočasné potvrdenie dočasnej žiadosti.
MESSAGE	umožňuje okamžitý prenos správy.
SUBSCRIBE	predznamenáva žiadosť NOTIFY.
NOTIFY	poskytuje informácie o zmene statusu nesúvisiaceho s konkrétnym dialógom.
PUBLISH	publikuje status udalosti serveru.
UPDATE	umožňuje klientovi upraviť parametre dialógu.
INFO	poskytuje signalizačné informácie počas dialógu.

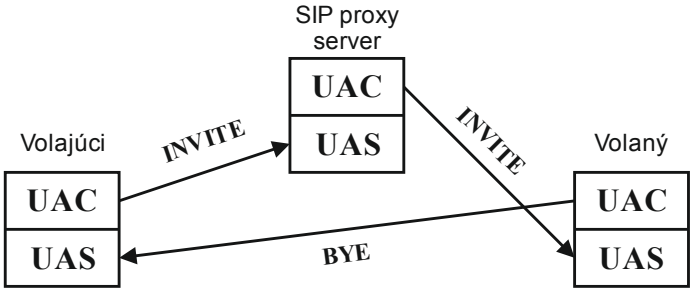
Odpovede sú označované číslami a sú rozdelené do 6 tried:

Číslo	Význam
1XX	informatívna odpoveď na žiadosť, ktorá bola prijatá, ale výsledok spracovania ešte nie je známy. Účelom je predísť opätovnému zasielaniu žiadosti.
2XX	pozitívna konečná odpoveď značiaca, že žiadosť bola úspešne spracovaná.
3XX	konečná odpoveď slúžiaca k presmerovaniu. Udáva novú polohu užívateľa alebo alternatívnu službu, ktorá má byť použitá.
4XX	konečná negatívna odpoveď, žiadosť nebola spracovaní chybou odosielateľa.
5XX	opäť konečná negatívna odpoveď, avšak v tomto prípade je chyba na strane prijímateľa.
6XX	globálna chyba, žiadosť nespracuje žiaden server.

1.2.2 Prvky architektúry protokolu SIP

Základnými prvkami sú:

- užívateľský agent (UA),
- proxy, registrar a redirect server.



Obrázok 1.2: *Prvky protokolu SIP*

UA je koncový bod siete, používaný k vzájomnému spojeniu. Jedná sa teda o koncový terminál, vo forme hardwarového či softwarového SIP telefónu, prípadne PSTN brány, IVR systému, atď. Každý užívateľský agent obsahuje User Agent Client (UAC), ktorého úlohou je odosielať požiadavky a prijímať odpovede a User Agent Server (UAS), ktorý naopak požiadavky prijíma a odosiela odpovede.

Špeciálnym prípadom užívateľského agenta je Back-to-Back User Agent (B2BUA), nachádzajúcim sa niekedy medzi koncovými bodmi. Na rozdiel od klasického proxy serveru správy nepreposiela, ale vytvára nové, a to v smere k obom účastníkom. Koncový terminál nezistí rozdiel, avšak B2BUA má tak za cenu menšieho počtu obslúžených žiadostí oveľa viac možností.

Koncové terminály odosielajú správy na proxy servery. Tie zaisťujú smerovanie žiadosti, v závislosti na aktuálnej pozícii adresáta, autentizáciu a veľa ďalších iných funkcií. Existujú dva druhy SIP proxy serverov:

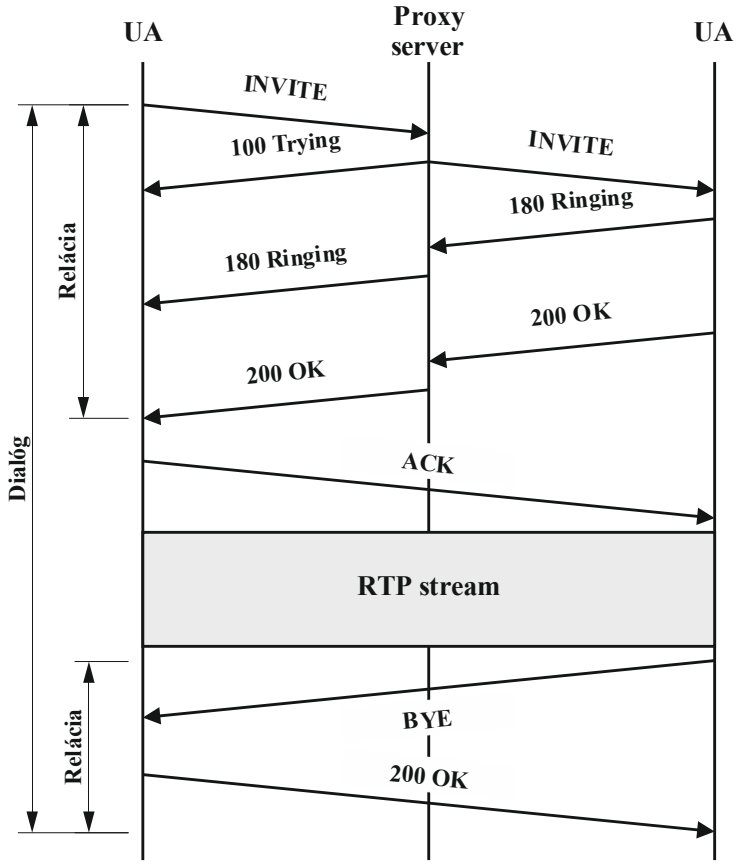
- stavový,
- bezstavový.

Bezstavový SIP proxy server len jednoducho preposiela správy, nezávisle na ich väzbách. Výhodou bezstavových proxy serverov je teda zníženie záťaže, avšak takýto server nedokáže detegovať opakovanie správ, ani využívať pokročilé smerovanie.

Stavový SIP proxy server si vytvára záznamy o prebiehajúcich transakciách. Nakoľko sa informácia o stave udržuje po celú dobu trvania transakcie, a tá môže byť pomerne dlhá, je tým výkon veľmi limitovaný. Stavové SIP proxy servery sa ďalej delia na transakčné stavové proxy, ktoré udržujú informáciu o stave transakcie od prijatia žiadosti, až po odoslanie konečnej odpovede a hovorové stavové proxy, sledujúce stav celého dialógu, od prvej žiadosti INVITE, po ukončenie pomocou BYE.

Transakcia je sekvencia SIP správ, odosielaných medzi SIP prvkami. Tvorí ju jedna žiadosť a všetky odpovede vztiahnuté k tejto žiadosti. Tzn. jednu alebo viac informatívnych odpovedí, a takisto jednu alebo viac konečných odpovedí.

Dialóg je množina správ, ktoré majú zhodné pole Call-ID, From a To. Vnútri dialógu sa môže nachádzať aj niekoľko transakcií, avšak aktívna môže byť vždy len jedna (obr. 1.3).



Obrázok 1.3: *Rozdiel medzi reláciou a dialógom*

Registrar servery prijímajú od užívateľov žiadosti na registráciu, vďaka čomu majú vo svojej lokalizačnej databáze informácie s ich súčasnou IP adresou, portom a užívateľským menom.

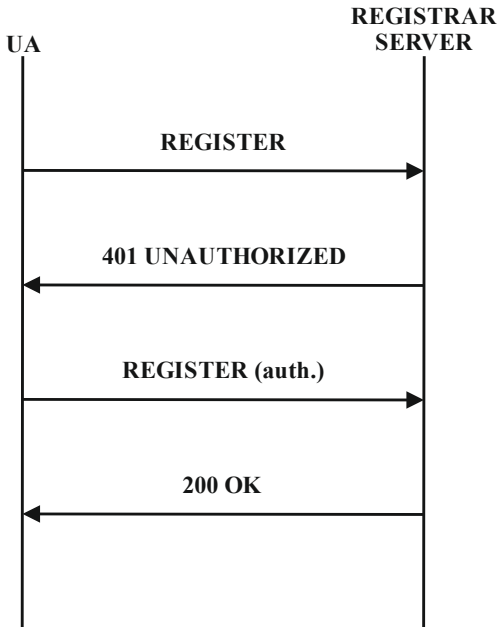
Redirect servery prijímajú požiadavky a pomocou lokalizačnej databázy vytvorenej registrar serverom vyhľadáva príjemcov. Odosielateľ tak od redirect serveru prijme zoznam s aktuálnou pozíciou hľadaného užívateľa a všetky ďalšie žiadosti smeruje priamo naňho. [13]

1.2.3 **Adresácia**

SIP využíva k adresovaniu domény. Každý prvok je identifikovaný pomocou SIP URI (Uniform Resource Identifier). Skladá sa z dvoch častí a má tvar sip:uzivatel@hostitel. Jednotlivé časti sú oddelené znakom @. V časti uzivatel je zadané meno alebo telefónne číslo a v časti hostitel doménové meno alebo IP adresa. SIP URI môže obsahovať aj ďalšie parametre ako napr. číslo portu (predvolene 5060) atď. V záujem zvýšenia bezpečnosti je dokonca možné vytvoriť kódované pripojenie, tzv. Secure SIP, jednoduchou zmenou prefixu zo „sip“ na „sips“. [15]

1.2.4 Registrácia a autentizácia

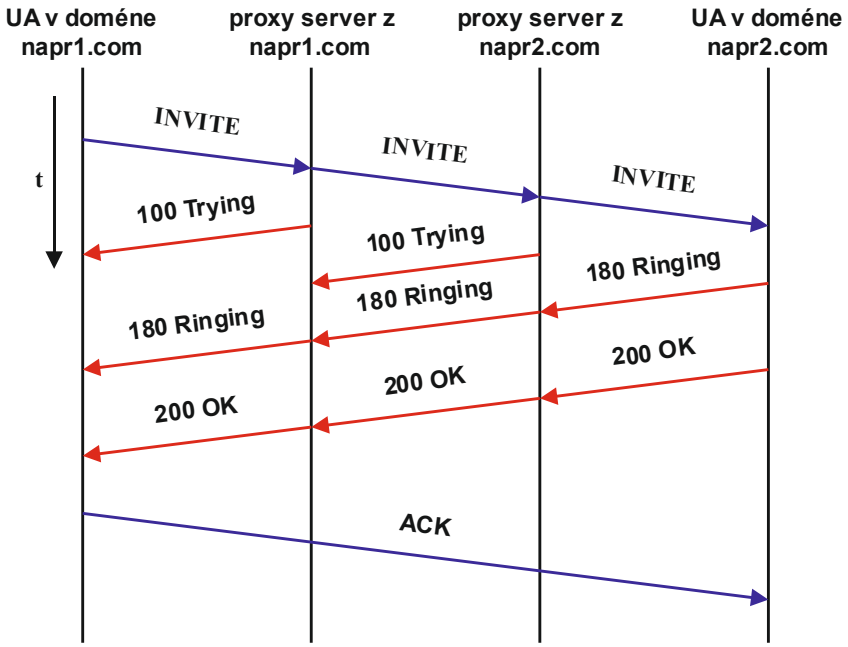
Ak chce byť užívateľ dostupný, musí sa zaregistrovať na registrar serveri. Registrácia sa skladá zo žiadosti REGISTER a odpovede 200 OK, zaslanou registrar serverom v prípade úspechu (obr. 1.4). V prípade, že je registrácia overovaná, dostáva užívateľ najskôr negatívnu odpoveď 401 UNAUTHORIZED alebo 407 PROXY AUTHENTICATION REQUIRED. Rozlišuje sa teda v autentizácii medzi dvoma užívateľmi a medzi proxy serverom a užívateľom. [15]



Obrázok 1.4: *Registrácia užívateľského agenta k registračnému serveru*

1.2.5 Smerovanie žiadosti o vytvorenie spojenia

Existujú dva možné scenáre smerovania žiadosti o vytvorenie spojenia. V prvom prípade, pozná odosielateľ presnú lokalizáciu adresáta a INVITE žiadosť odosiela priamo jemu. V druhom prípade je žiadosť najskôr odosielaná na proxy server, ktorý ju posiela ďalej až do momentu, než je adresát lokalizovaný (obr. 1.5). Nakoľko následná odpoveď volaného aj pôvodná žiadosť volajúceho obsahujú pole Contact, obidvaja UA poznajú presnú lokalizáciu a ďalšia výmena už prebieha len medzi UA bez účasti proxy serverov.



Obrázok 1.5: *Príklad smerovania žiadosti v SIPe*

1.2.6 Zabezpečenie v SIPe

Protokol SIP nevyužíva žiadnu formu zabezpečenia a je prenášaný ako otvorený text. K zachyteniu a prečítaniu komunikácie nie je nutné použiť žiadny pokročilý analyzátor. Princíp fungovania komunikácie je podobný s HTTP, čo znamená, že je možné využiť tie isté bezpečnostné mechanizmy ako pri HTTP.

Prvým zo spomínaných mechanizmov je HTTP Basic Autentication. Poskytuje len autentizáciu pomocou zdieľaného hesla, dáta zostávajú nešifrované a nie je ani zaručená ich integrita.

HTTP Digest Autentication vychádza z HTTP Basic Autentication, takže ani tento mechanizmus neposkytuje šifrovanie dát a nezaručuje integritu. Zmenou je, že heslo je pri prenose zašifrované pomocou hashovacej funkcie MD5 alebo SHA.

Secure Multipurpose Internet Mail Extension zaisťuje šifrovanie aj integritu. Šifrovanie je možné dosiahnuť dvoma metódami. Prvá využíva šifrovanie a digitálne podpisy, druhá infraštruktúru verejných kľúčov a symetrické šifry (DES, 3DES, AES).

SIPS URI využíva k autentizácii infraštruktúru verejných kľúčov. Tým sa zmení prefix v SIP URI zo „sip:“ na „sips:“. Šifrovanie a integrita sú zaručené protokolom TLS (Transfer Layer Security), ktorý využíva symetrické šifry, asymetrické šifry aj hashovacie funkcie. V tomto prípade sa však na prenos využíva protokol TCP a nie UDP. [15]

1.2.7 SDP

Session Description Protocol popisuje multimediálne relácie za účelom oznámenia, pozvania alebo iných foriem vytvorenia multimediálnej relácie. Používa sa pri propagácii multimediálnych konferencií a poskytuje všetky nevyhnutné informácie k pripojeniu sa. Množstvo SDP správ je odosielaných prostredníctvom SAP (Session Announcement Protokol) protokolu ako multicastové UDP pakety so SAP hlavičkou a SDP informáciami ako nákladom. [19]

SDP tvoria riadky s textom v tvare „typ=hodnota“ rozdelené do troch skupín:

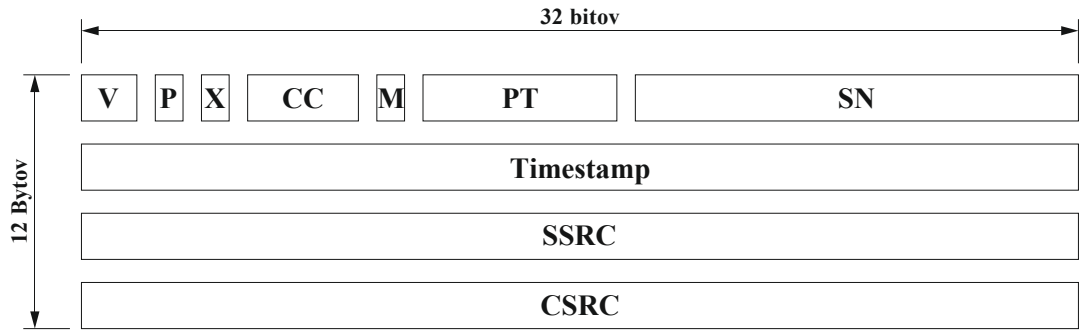
- popis relácie,
- popis časovania,
- popis médií.

1.3 RTP

Aj napriek označeniu za protokol aplikačnej vrstvy je Real Time Protokol oveľa vhodnejšie priradiť nad UDP na transportnú vrstvu, pretože má s transportnými protokolmi množstvo spoločných znakov v obsahu a účelu polí hlavičiek. Ďalšie vlastnosti, ktoré má RTP spoločné s protokolmi transportnej vrstvy sú zoraďovanie zaslaných paketov pomocou sekvenčných čísiel, časové značenie, multiplexing a demultipexing. [13]

V RFC sú definované tri verzie Real Time Protokolu:

- RFC 1889 z roku 1996 obsahuje pôvodný Transport Protocol for Real-Time Applications,
- RFC 3550 z roku 2003 vylepšuje RFC 1889 o dohľad nad RTP tokom,
- RFC 3711 z roku 2004 pridáva zabezpečenie RTP, tzv. Secure Real-time Transport Protocol, a možnosť zmenšenia veľkosti hlavičky zo 40 bytov na 2-3 byty (compressed Real-time Transport Protocol, cRTP). Kompresia hlavičky musí byť podporovaná na obidvoch stranách a tak sa využíva takmer výhradne len v spojoch bod-bod.



Obrázok 1.6: *Hlavička RTP datagramu*

Hlavička RTP datagramu má veľkosť 12 bytov tvorí ju niekoľko polí (viď. obr):

- V – verzia RTP,
- P – doplnenie, v prípade použitia paket obsahuje jeden či viac Bytov nereprezentujúcich užitočné informácie,
- X – rozširujúci bit, v prípade použitia po pevnom záhlaví nasleduje rozšírené záhlavie s definovanou štruktúrou,
- CC – číslo identifikátoru CRSC,
- M – značka, môže mať rôzne využitie,
- PT – označenie formátu a významu užitočnej informácie nesenej RTP,
- SN – sekvenčné číslo využívané pri opätovnom zoraďovaní paketov v prijímači,
- Timestamp – časová značka prvého bytu v RTP pakete,
- SSRC – identifikátor synchronizačného zdroja, je volený náhodne a v rámci jednej relácie nesmú mať dva synchronizačné zdroje zhodné SSRC,
- CSRC – identifikátor prispievajúceho zdroja.

1.4 SRTP

Z dôvodu, že RTP rovnako ako SIP neobsahuje žiadne zabezpečovacie mechanizmy, bol v roku 2004 v RFC 3711 definovaný protokol Secure Real-time Transport Protocol. Ten vychádza zo RTP, ale umožňuje šifrovanie užitočnej informácie a zaistenie integrity hlavičky autentizačnou značkou. Autentizačná značka je výstupom hashovacieho algoritmu HMAC-SHA-1 so vstupnými premennými užitočná informácia RTP a autentizačný kľúč [15]. SRTP šifruje v dvoch definovaných režimoch:

- AES-CTR,
- AES-f8.

1.5 ZRTP

Dopĺňa protokol SRTP o mechanizmus počiatočnej výmeny kľúčov a o ochranu proti útokom typu MITM. ZRTP využíva pri vytváraní kľúčov Diffie-Hellmanov algoritmus. To znamená, že komunikujúce strany si vymieňajú reťazce, z ktorých si odvodia tajný šifrovací kľúč a ten sa teda vôbec neprenáša. Útočník by musel disponovať obrovským výpočtovým výkonom a riešiť diskrétny logaritmus s obrovskými číslami. Detekcia útokov MITM je zaistená pomocou metód Short Authentication String (SAS) a Retained Secrets (RS). ZRTP sa skladá z troch fáz:

- detegovanie podpory ZRTP účastníkmi,
- tvorba kľúčov pomocou DH algoritmu,
- prechod do SRTP módu.

1.6 RTCP

Real Time Control Protokol zbiera údaje o prenose prostredníctvom RTP, štatistiky o množstve odoslaných dát, počte odoslaných a stratených paketov, rozptyle omeškania [13]. RFC 3550 obsahuje 5 typov správ:

- Sender Report – štatistiky od odosielateľov,
- Receiver Report – štatistiky od príjemcov,
- Source Description – informácie o odosielateľoch RTP dát,
- BYE – ukončenie spojenia,
- APP – má význam podľa aplikácie.

2 Typy útokov

2.1 Stopovanie

Množstvo spoločností by bolo prekvapených pri zistení, koľko citlivých informácií je o nich verejne dostupných. UC siete však už nie sú len doménou obrovských spoločností. Rozšírili sa aj v stredne veľkých, či malých podnikoch a je jasné, že tým narastá aj počet potenciálnych cieľov pre útočníka.

Stopovanie je jednou z najdôležitejších častí prvotného výskumu, akým spôsobom by bolo možné získať prístup do cieľovej siete. „Najlepšou“ vecou na stopovaní je, že útočník sa cieľovej siete nijako nedotýka, takže na získanie potrebných informácii má toľko času, koľko potrebuje. Veď je to v jeho samotnom záujme, získať pred útokom čo najviac informácií. Veľakrát je najjednoduchším spôsobom, ako kompromitovať UC systém, využiť k tomu podpornú infraštruktúru, napr. webový server alebo hlasovú bránu. [2]

2.1.1 Prehľadávanie webových stránok

Webové stránky spoločností často poskytujú množstvo informácií. A hoci sú považované za neškodné, pretože ich cieľom je predstaviť spoločnosť návštevníkom, sú to vrátka umožňujúce prístup sociálnemu inžinierovi do spoločnosti [3]. Jedná sa najmä o:

- organizačnú štruktúru a umiestnenie spoločnosti,
- technickú podporu,
- ponuky práce,
- telefónne čísla.

2.1.2 Google hacking

Je príkladom stopovania, využívajúci potenciál internetových vyhľadávacích nástrojov v odhaľovaní najväčších bezpečnostných rizík. Útočník len jednoducho využije pokročilé funkcie služieb ponúkaných Googlom. Dokonca sa ponúka možnosť využiť aj iné vyhľadávače ako napr. Yahoo! alebo Bing a dopátrať sa tak k odlišným výsledkom. Cenné výsledky môže priniesť hlavne zameranie sa na tieto 4 kategórie vo vyhľadávaní:

- tlačové správy a štúdie,
- životopisy,
- poštové správy a užívateľské príspevky,
- prihlasovanie UC založené na webe.

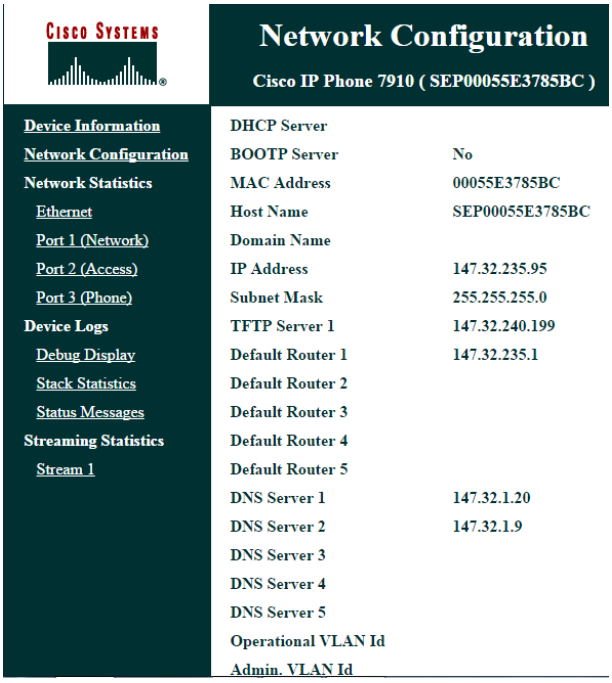
Niektorí predajcovia UC zariadení a služieb vydávajú v prípadoch zisku veľkej zákazky tlačové správy. Občas tieto tlačové správy obsahujú štúdie s popisom konkrétnych produktov a ich verzií, ktoré boli zákazníkovi poskytnuté.

Obdobne ako popisy v ponukách práce obsahujú aj životopisy množstvo potenciálne užitočných informácií. Stačí niekoľko vhodne zvolených kľúčových slov vo vyhľadávači a je

možné dostať sa k informáciám ako napr.: „Viac ako 5 ročné skúsenosti v návrhu, vývoji a správe Cisco sieťovej infraštruktúry obsahujúcej dátové, hlasové a bezdrôtové technológie.“

Užívateľské fóra a technické správy sú neoceniteľným zdrojom informácií pre začínajúcich správcov sietí snažiacich sa dozvedieť viac o UC technológiách. Hoci s najlepšími úmyslami, hnaný snahou získať pomoc od komunity, tak často poskytnú až priveľa informácií. Opačným prípadom sú skúsení správcovia zdieľajúci vlastné konfiguračné súbory, pomocou ktorých chcú menej zdatným kolegom zjednodušiť náročné nastavovanie niektorých funkcií.

Väčšina UC zariadení disponuje webovým rozhraním umožňujúcim sieťový manažment a zmenu osobných nastavení. Takéto systémy by nemali byť voľne dostupné prostredníctvom internetu a to nielen z dôvodov možnosti napadnutia útokom prelomenia hesla hrubou silou, ale aj pretože tak môžu odhaliť chybu v zabezpečení webového servera. Avšak stačí zadať do Googlu jednoduchú frázu: „inurl:"NetworkConfiguration" cisco“, a zobrazí sa hneď niekoľko odkazov [3]. Po otvorení jedného z nich sa zobrazí napr.:



Obrázok 2.1: *Webové rozhranie s informáciami o telefóne*

2.2 Skenovanie siete

Ďalším krokom je overiť každú IP adresu z rozsahu získaného stopovaním, zistiť „živé“ systémy a identifikovať na nich bežiace služby pre potreby sieťovo založených útokov alebo rozsahy používaných telefónnych čísiel pre potreby útokov na aplikačnej úrovni. Skenovanie sa od stopovania odlišuje aj v tom, že je ľahko odhaliteľné.

Prostredie UC netvoria výhradne telefóny a servery. A pretože dostupnosť a bezpečnosť UC sietí je veľmi závislá na podpornej infraštruktúre, bolo by od útočníka veľmi hlúpe, keby zameral len na zariadenia, na ktorých bežia samotné služby. Preto sa cieľom útoku často stávajú

smerovače, VPN brány, hlasové brány, TFTP servery, DNS servery, DHCP servery, firewally a ďalšie sieťové zariadenia.

Ak by bol útočník schopný nájsť a odstaviť TFTP server, môžu niektoré telefóny počas bootovania pri sťahovaní konfiguračného súboru zlyhať. Preťažením DHCP serveru by sa telefón, neúspešne sa pokúšajúci získať IP adresu stal nepoužiteľným. Na konci takéhoto skenovania by sa mal útočník stať schopným identifikovať jadro sieťovej infraštruktúry a všetky sieťovo dostupné UC systémy. [2]

2.2.1 ICMP ping

ICMP ping je jednoduchý spôsob vyhľadávania aktívnych strojov. Skenovanie pozostáva z odoslania paketu ICMP ECHO REQUEST na IP adresu. V prípade nezablokovania správy smerovačom alebo firewallom, väčšina zariadení odpovie paketom ICMP ECHO REPLY. [4]

2.2.2 TCP ping

V prípade, že je akákoľvek ICMP premávka blokovaná smerovačom alebo firewallom cieľovej siete existuje ešte niekoľko ďalších spôsobov ako môže externý útočník detegovať aktívne stroje. To zahŕňa využitie počiatočného vytvárania spojenia v TCP/IP a značiek pritom používaných.

Jedna z týchto metód, zvaná TCP ping, odosiela pakety so značkou TCP SYN alebo ACK so známymi číslami portov na cieľové adresy. V prípade úspechu, tak stroj odpovie paketom s RST značkou. V tomto prípade je ale výhodnejšie používať ACK pakety, pretože niektoré bezstavové firewally blokujú SYN pakety. [4]

2.2.3 SNMP skenovanie

Ďalším zo spôsobov, ako odhaliť aktívne sieťové vybavenie je prostredníctvom Simple Network Management Protocolu (SNMP). SNMP je protokol aplikačnej vrstvy umožňujúci monitorovanie a správu sieťových zariadení. K dispozícii sú tri verzie SNMP:

- SNMP v1 (RFC 1067)
- SNMP v2 (RFC 1441 - 1452)
- SNMP v3 (RFC 3411 - 3418)

SNMP v1 je najširšie podporovaným protokolom používaným množstvom UC telefónov za účelom spätnej kompatibility. Medzi jednotlivými troma verziami je množstvo rozdielov, avšak najdôležitejším je, že SNMP v1 a v2 spoliehajú len na jednoduchú formu autentizácie, tzv. komunitné reťazce, čo je v podstate len nijako nechránené heslo. SNMP v3 využíva silnejšie šifrovanie ako napr. AES alebo 3DES. [2]

2.2.4 TCP SYN a UDP skenovanie portov

Pri TCP SYN skenovaní sa nevytvára úplné TCP spojenie. Odošle sa SYN správa ako pri vytáraní reálneho spojenia a potom sa čaká na odpoveď. V závislosti na odpovedi potom porty označuje ako:

Tabulka 1.1: *Označovanie portov pri TCP SYN skenovaní podľa prijatej odpovede*

Typ odpovede	Označenie portu
SYN/ACK, SYN	otvorený
RST	zatvorený
ICMP nedostupný (typ 3, kód 1, 2, 3, 9, 10, 13)	otvorený
-	filtrovaný

UDP skenovanie odosiela UPD paket na všetky cieľové porty. Na niektoré, zo známych portov posiela ako užitočnú informáciu špecifickú správu, v ostatných prípadoch je paket prázdny. Aj v tomto prípade je označenie portu závislé na odpovedi.

Tabulka 1.2: *Označovanie portov pri UDP skenovaní podľa prijatej odpovede*

Typ odpovede	Označenie portu
ICMP nedostupný (typ 3, kód 3)	zatvorený
ICMP nedostupný (typ 3, kód 1, 2, 9, 10, 13)	filtrovaný
UDP paket	otvorený
-	otvorený|filtrovaný

Význam označení portov:

Označenie	Význam
Otvorený	Aplikácia aktívne TCP spojenia a UDP pakety na danom porte.
Zatvorený	Zatvorený port je dostupný (prijíma a odpovedá na testovacie pakety), ale žiadna aplikácia na tomto porte nepočúva.
Filtrovaný	Nmap nedokáže určiť, či je port otvorený alebo nie, pretože firewall alebo pravidlá na smerovači filtrujú pakety.
Nefiltrovaný	Port je dostupný, no Nmap nedokáže určiť či je otvorený alebo zatvorený.
Otvorený|Filtrovaný	Nmap nedokáže určiť, či je port otvorený alebo filtrovaný.
Zatvorený|Filtrovaný	Nmap nedokáže určiť, či je port zatvorený alebo filtrovaný.
TCPWrapped	TCP Wrapper slúži ako firewall pre UNIXové servery a filtruje prichádzajúce pakety. Sleduje, či má daná entita autorizáciu na vytvorenie spojenia, v prípade, že nie, prístup zamietne.

2.2.5 Snímanie odtlačku prstov

Po preskenovaní TCP a UDP portov je užitočné určiť typ zariadenia, operačný systém a firmware. Aj keď niektoré otvorené porty môžu naznačiť použitý operačný systém, nikdy nie je na škodu si ďalšou technikou testovania hypotézu potvrdiť. Touto technikou je práve tzv. snímanie odtlačku prstov, ktoré sleduje odlišnosti v sieťových žiadostiach jednotlivých operačných systémov a firmwaru. [3]

2.3 DoS

Denial of Service je útok charakterizovaný ako pokus útočníka zabrániť užívateľom prístup k službe. DoS útok môže v podstate odstaviť počítač, sieť a v závislosti na charaktere podnikania dokonca aj celú organizáciu. Niektoré DoS útoky sú vykonávané s limitovanými zdrojmi proti rozsiahlym sofistikovaným sieťam. Tieto typy útokov sa nazývajú asymetrický útok.

DoS útoky prichádzajú v rôznych formách a sú zamerané na rôzne služby. Je možné ich rozdeliť do troch základných skupín:

- spotreba obmedzených alebo nenahraditeľných zdrojov,
- odstránenie alebo zmena konfigurácie,
- fyzická deštrukcia sieťových komponentov.

DoS útoky sú najčastejšie používané pri útoku na sieťovú konektivitu. Cieľom je zabrániť počítaču, prípadne celej sieti v komunikácii. Príkladom takéhoto útoku môže byť tzv. SYN flood. V tomto prípade útočník začne proces s vytváraním spojenia so strojom obete, avšak nikdy ho nedokončí úplne. Cieľový stroj tomuto spojeniu musí rezervovať časť

dostupných zdrojov, čím sa útočníkovi podarí aby cieľový stroj nemohol vytvoriť žiadne iné, hoci aj legitímne spojenie.

Útočník sa môže pokúsiť zabrať celú šírku pásma siete generovaním veľkého množstva paketov smerovaných do siete. Zvyčajne sa jedná o pakety ICMP ECHO odosielané z jedného, prípadne viacerých počítačov, čím sa výsledný efekt len posilní. Tento typ útoku môže konzumovať aj iné zdroje, napr. zahltiť mailovú schránku množstvom správ alebo generovať chybové správy, ktoré sa musia logovať.

Odstránenie alebo zmena konfigurácie obvykle znamená zmenu konfigurácie počítača, prípadne iného zariadenia, následkom čoho zariadenie nemusí pracovať správne alebo môže prestať pracovať úplne.

Cieľom fyzickej deštrukcie sieťových komponentov je fyzické zabezpečenie. Útočníci majú záujem nielen o počítače, smerovače, ale aj o napájacie a chladiace stanice a iné kritické komponenty siete. [17]

2.3.1 Útok záplavou UDP

Je preferovaným typom záplavového útoku pretože UDP zdrojová adresa je jednoducho sfalšovateľná. Zaplavovanie umožňuje útočníkovi manipulovanie s dôveryhodnosťou v rámci organizácie prekonávaním firewallov a ostatných filtračných zariadení (napr. zručným zamaskovaním za DNS odpoveď na UDP porte 53).

Takmer všetky zariadenia s podporou SIPu podporujú aj UDP, čo z nich činí vynikajúci počiatočný bod útoku. Množstvo UC zariadení a operačných systémov môže byť odstavených záplavou UDP paketov na porte 5060, prípadne aj inom náhodnom porte [2].

2.3.2 Útok záplavou TCP SYN

Takýto útok zvyčajne začína množstva SYN paketov s podvrhnutými zdrojovými IP adresami. Obeť na jednotlivé žiadosti odpovedá zasielaním SYN-ACK. Podstatou útoku však je, že finálna odpoveď ACK už nikdy nepríde a obeti sa tak rýchlo zaplní tabuľka s pripojeniami, čím dôjde k vyčerpaniu dostupných výpočtových zdrojov spracovávaním týchto žiadostí. Výsledkom je, že cieľový stroj nie je schopný rozlíšiť medzi skutočnými a falošnými SYN paketmi [3].

2.3.3 Manipulácia s kvalitou služby cieleným zaplavovaním

Oveľa pokročilejší typ záplavového útoku ovplyvňujúci mechanizmy zaručujúce kvalitu určitých služieb v rámci siete s cieľom poškodiť UC aplikácie. Vychádza z predpokladu, že technológie zaisťujúce QoS uprednostňujú premávku RTP pred všetkým ostatným a tak by bol vnútorný záplavový útok neefektívny. Avšak v prípade, že by dokázal zaplaviť telefón, proxy server alebo pobočkovú ústredňu legitímne vyzerajúcou premávkou, QoS mechanizmy by neboli schopné rozoznať, ktorá konverzácia je správna a ktorá podvodná. [2]

2.3.4 Paketová fragmentácia

Hoci je paketová fragmentácia pomerne starý útok, zostáva významným. „Správnou“ fragmentáciou TCP aj UDP paketov je možné dosiahnuť, že množstvo UC zariadení a operačných systémov sa stane nepoužiteľnými. Existuje množstvo verzií, medzi najpopulárnejšie patria napr.: Scapy, Metasploit, teardrop, opentear, nestea, boink a The ping of death. [3]

2.3.5 Významné slabiny operačných systémov a firmwaru

Ďalšia významná kategória DoS útokov na UC infraštruktúru využíva bezpečnostné medzery v aplikáciách alebo operačných systémoch, čo vo výsledku môže viesť ku zlyhaniu alebo konzumácií voľných zdrojov. Napr. nová slabina v Linuxe môže ovplyvniť Asterisk bežiaci na ňom. [2]

2.3.6 Vyčerpanie DHCP

Množstvo UC telefónov v predvolenom nastavení žiada pri každom zapnutí o dynamické pridelenie IP adresy. Ak nie je počas načítavania dostupný DHCP server alebo sú už všetky adresy rezervované, telefón nie je v sieti možné použiť. Existuje niekoľko nástrojov, ktoré je možné použiť na vyčerpanie voľných adries DHCP serveru, napr. Yersinia. Yersinia bola navrhnutá k testovaniu niektorých slabín rôznych sieťových protokolov. [1]

2.4 Krádež registrácie

Odpovedá scenáru, kedy nahradí legitímnu registráciu falošnou. Pod takýmto útokom je možné si predstaviť presmerovanie hovorov na neexistujúce zariadenie, presmerovanie prichádzajúcich hovorov na iný SIPový koncový bod alebo presmerovanie hovorov na podvodnú aplikáciu.

Útočník môže napodobňovať správanie užívateľa alebo použiť útok aplikačnej úrovne zvaný Man in the Middle, ktorým sa dostane medzi dve komunikujúce strany. V tejto veľmi výhodnej pozícii potom jednoducho zachytáva alebo dokonca upravuje prechádzajúcu komunikáciu.

Krádež registrácie sa používa aj k presmerovaniu hovoru na podvodnú aplikáciu, ktorá sa potom snaží od volajúcich získať citlivé informácie. Takáto aplikácia predstiera správanie hlasovej schránky odpovedaním na hovory a prehrávaním vygenerovaných správ. Množstvo užívateľov si ani neuvedomí, že sa nejedná o hlasový systém spoločnosti. Sofistikovaný útočník si môže jednoducho nahrať odpovede z aktuálneho serveru a urobiť tak útok ešte ťažšie detekovateľný. [3]

2.5 MITM

V tradičnom Man in the Middle útoku útočník umiestnený medzi dvoma komunikujúcimi stranami, bez ich vedomia odpočúva a prípadne aj pozmeňuje prechádzajúce dáta. Toho docieli napr. podvrhnutím SIP proxy alebo umiestnením sa medzi užívateľa a regulárnu SIP proxy. Táto pozícia útočníkovi umožňuje napr.:

- odpočúvať konverzáciu,
- spustiť DoS útok,
- pozmeniť konverzáciu vynechaním, vložením alebo prehrávaním dát,
- presmerovať hovor na iného príjemcu.

V rozsiahlej infraštruktúre s UC podporou je množstvo ďalších vecí, ktoré útočník dokáže vďaka MITM útoku. V prípade, že sa útočníkovi podarí dostať medzi užívateľa UC a dôležitý server (napr. TFTP, DNS) tak môže:

- podvrhávať TFTP, DNS a DHCP,
- presmerovať ICMP,
- manipulovať so smerovaním.

Každý z týchto útokov sa dá použiť k záznamu alebo narušeniu hovorov v spoločnosti. Množstvo ovplyvnených užívateľov závisí od typu útoku. Niektoré z nich dokážu postihnúť hovory hŕby užívateľov, čo môže mať dôsledky obzvlášť, ak je útok smerovaný na zákazníkov. Väčšina útokov však vyžaduje priamy prístup do vnútornej siete a výsledný úspech je závislý na úrovni zabezpečenia. [14]

2.5.1 Otrávenie ARP

Otrávenie ARP je jednou z najobľúbenejších techník prevádzkovania MITM útokov, kde odpočúvanie je len jedným z potenciálnych dopadov. Otrávenie ARP spolieha na to, že mnoho operačných systémov prijme alebo nahradí ARP záznamy, bez závislosti na prvotnom odoslaní žiadosti. Útočník potom začne vydávať svoju MAC adresu za adresu druhého počítača alebo kriticky dôležitého servera a smerovaním premávky pôvodnému príjemcovi utají svoju prítomnosť. [1]

2.6 SPIT

Spam over Internet Telephony je definovaný ako nežiadúce hromadné volanie vedúce k vytváraniu spojení prenášajúcich do telefónov alebo hlasových terminálov zvuk, obraz alebo video. Existuje niekoľko druhov SPIT, napr. reklama, telemarketing či telefonický dotazník. V porovnaní s emailovým spamom môže byť nárok na sieťové zdroje až desaťnásobný. SPIT je aj viac otravný, pretože s každým prichádzajúcim spamom zazvoní telefón nezávisle na aktuálnom čase či vykonávanej činnosti. Použitie VoIP miesto klasickým PSTN sietí spamerom zjednodušilo prácu, pretože môžu používať lacné automatizované nástroje. Náklady na volanie prostredníctvom VoIP sú o tri rády nižšie ako pri PSTN sieťach.

SPIT má za cieľ vytvoriť tak veľa komunikačných spojení s obeťami útoku, ako je to len možné. Takýto útok sa skladá z troch krokov. Prvým krokom je systematické zhromažďovanie kontaktov na osoby. Druhým krokom je vytváranie spojení a tretím odoslanie správy. [18]

2.6.1 **Manuálne obťažujúce hovory**

Žartovné, otravné a obťažujúce hovory existujú už veľmi dlhú dobu. Problém sa čiastočne zlepšil možnosťou vidieť číslo alebo ID volajúceho aj pred prijatím hovoru. Ale nie je veľmi zložité volať z anonymného alebo podvrhnutého čísla, a v prípade, že zámerom hovoru je obťažovať alebo verbálne napadnúť obeť, môže falošné číslo pomôcť oklamať a tak primäť obeť zodvihnúť hovor. V prípade, že nedôjde k ohrozeniu obete alebo k ovplyvňovaniu obchodných operácií, sú tieto hovory legálne. Takýto útok je však oveľa väčším problémom pre jednotlivca ako pre veľkú spoločnosť. [18]

2.6.2 **Výhražné hovory**

Bomba a iné bezpečnostné hrozby sú špecifickým typom obťažujúcich hovorov. Môžu byť veľmi nebezpečné, pretože väčšina organizácií prístupných verejnosti má pravidlá prikazujúce evakuovať celú oblasť v prípade prijatia vyhrážky. Množstvo z nich pochádza od zamestnancov snažiacich sa dostať deň voľna alebo od nespokojných bývalých zamestnancov túžiacich po pomste. Hoci je pomerne jednoduché rozpoznať útočníkov hlas, sú tieto útoky dosť časté. Okrem rušivého a finančného dopadu však môžu vyústiť v chaos a následnú fyzickú ujmu na zdraví. [2]

2.6.3 **Hlasový SPAM**

Jednoduchým spôsobom ako generovať hlasový SPAM je využiť komerčné služby. Jedná sa o legitímne služby najčastejšie používané k doručovaniu telemarketingových hovorov, reklám, politických propagácií, ale využitie môžu nájsť aj pri rozosielaní hlasového SPAMu. Ich nevýhodou je, že nie sú bezplatné a ich prevádzkovatelia by nemuseli súhlasiť s takým zneužitím ich služieb. [3]

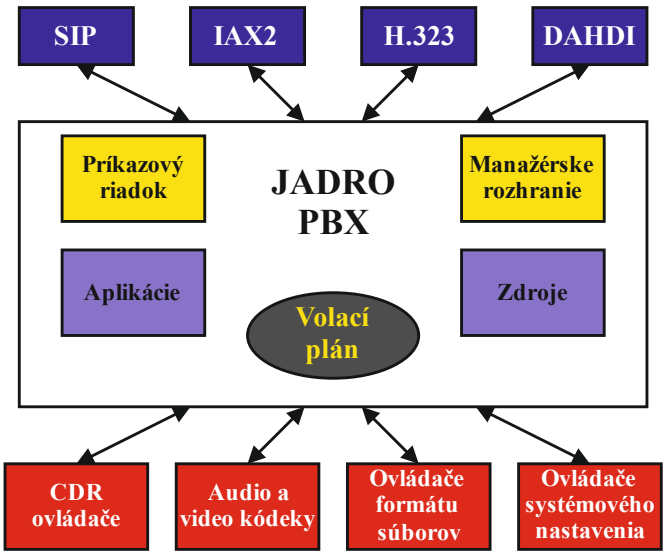
3 Testovaný a testovací software

3.1 Asterisk

Prvý nápad sa zrodil u zakladateľa Marka Spencera v roku 1999. Množstvo spoločností sa snažilo ušetriť peniaze tým, že nahradia nákladné proprietárne operačné systémy voľne dostupným Linuxom. Problémom však bol nedostatok podpory a práve túto medzeru sa Mark rozhodol zaplniť vytvorením Linuxových podporných služieb. LPS ponúkali za drobný poplatok odbornú pomoc s Linuxom. Po niekoľkých mesiacoch sa tak z malej kancelárie a tímu linuxových odborníkov stala veľká spoločnosť založená na telefónnom systéme doručujúcom hovory naprieč celým podporným tímom so ziskom viac ako $50 000.

V spolupráci s Adtranom, výrobcom komunikačných a sieťových zariadení sa rozhodol napísať vlastný telefónny systém. Krátko na to priniesla spolupráca ovocie vo forme prvotného kódu jadra Asterisku. Funkčný prototyp bol okamžite vydaný na internet pod licenciou GPL (rovnakú používal aj Linux). Asterisk zožal veľký úspech a stovky vývojárov z celého sveta začali pridávať množstvo nových funkcií a súčastí.

Komunikačné aplikácie Asterisku sú založené na skriptoch hovorových plánov, konfiguračných súboroch, zvukových nahrávkach, databázach, viď obr. 3.1:



Obrázok 3.1: *Architektúra Asterisku*

Po viac ako desaťročí vývoja tak Asterisk môže Asterisk zastávať rolu:

- VoIP brány,
- pobočkovej ústredne,
- softwarovej ústredne,
- konferenčného serveru,
- packet voice serveru,
- voicemail služby,
- interaktívneho hlasového sprievodcu,
- prekladača čísiel,
- šifrovacieho prvku, atď.

Podporuje signalizačné protokoly:

- SIP
- H.323
- IAX2
- MGCP
- SCCP
- Nortel unistim

A taktiež hneď niekoľko kódekov:

- G.711 ulaw,
- G.711 alaw,
- G.722,
- G.723.1,
- G.726,
- G.729,
- GSM,
- iLBC,
- LPC10,
- Speex.

V súčasnosti existuje niekoľko podporovaných verzií Asterisku. Každá vydaná verzia je podporovaná limitovaný čas. Na začiatku sú vydávané záplaty na všetky nahlásené chyby. V momente, kedy je verzia označená za zastralú, sú vydávané len záplaty týkajúce sa bezpečnostných rizík. V poslednej fáze Koniec života už nie sú vydávané žiadne záplaty. [12]

Rozlišuje sa medzi dvoma verziami. Long Term Support má plnú podporu počas štyroch rokov od vydania plus jeden rok navyše na odstraňovanie bezpečnostných rizík. Štandardné vydanie je plne podporované rok a druhý rok len v rámci vydávania bezpečnostných záplat.

Tabulka 1.3: *Verzie Asterisku*

Verzia	Typ verzie	Dátum vydania	Len bezpečnostné záplaty	Koniec života
1.2.X		21.11.2005	7.8.2007	21.11.2010
1.4.X	LTS	23.12.2006	21.4.2011	21.4.2012
1.6.0.X	Štandardná	1.10.2008	1.5.2010	1.10.2010
1.6.1.X	Štandardná	27.4.2009	1.5.2010	27.4.2011
1.6.2.X	Štandardná	18.12.2009	21.4.2011	21.4.2012
1.8.X	LTS	21.10.2010	21.10.2014	21.10.2015
10.X	Štandardná	15.12.2011	15.12.2012	15.12.2013
11.X	LTS	25.10.2012	25.10.2016	25.10.2017
12.X	Štandardná	20.12.2013	20.14.2014	20.12.2015
13.X	LTS	24.10.2014	24.10.2018	24.10.2019

3.2 Nessus

Nessus je vynikajúci nástroj navrhnutý k automatickému testovaniu a vyhľadávaniu známych bezpečnostných problémov. Zvyčajne je to skupina hackerov alebo spoločnosti zaoberajúce sa bezpečnosťou, ktoré odhalia určitý spôsob, ktorým je možné obísť zabezpečenie softwaru. Tento objav môže byť úplne náhodný, ale aj výsledkom cieleného výskumu. Nessus je navrhnutý tak, aby pomohol odhaliť a vyriešiť tieto problémy ešte pred tým, ako ich zneužije potenciálny útočník. Ponúka množstvo funkcií a aj napriek komplexnosti existuje množstvo manuálov, ktoré môžu byť začínajúcim užívateľom veľmi nápomocné.

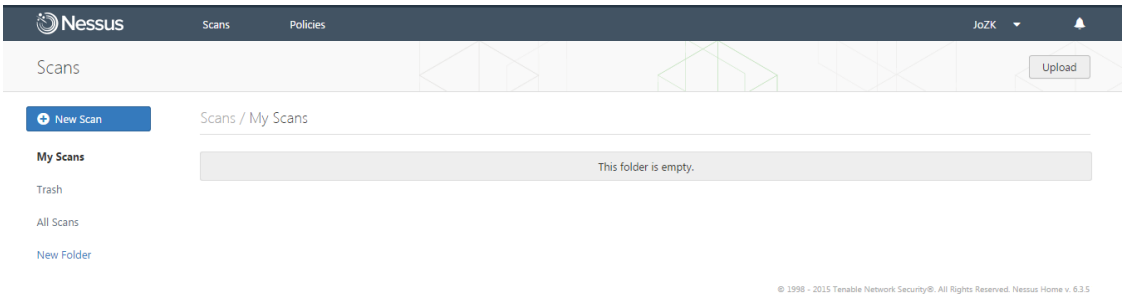
Jednou z najdôležitejších funkcií je technológia klient server. Server môže byť umiestnený na rôzne miesta v sieti, čím sprístupní viacero uhlov pohľadu na vykonávané testy. Centrálny klient, prípadne viacero klientov môže ovládať všetky servery. Časť serveru môže bežať na ľubovoľnom Unixovom, MAC OS alebo IBM/AIX operačnom systéme, avšak v prípade Unixu je inštalácia najjednoduchšia. Klient je dostupný pre Windowsové aj Unixové platformy. Server vykonáva testovanie, zatiaľ čo klient sprostredkúva hlásenia a sprístupňuje konfiguráciu. [16]

3.2.1 Prostredie Nessusu

Do aplikácie sa prihlasuje cez webový prehliadač zadaním adresy: „https://localhost:8834“. Po úspešnom prihlásení sa zobrazí úvodné menu (obr. 3.2). To obsahuje dve hlavé sekcie:

- Scans,
- Policies.

V sekcii Scans je možné vytvárať, upravovať alebo odstraňovať profily skenovaní. V sekcii Policies je možné vytvárať, upravovať alebo odstraňovať pravidlá pre profily skenovaní. Nastavenia užívateľského profilu a všeobecné nastavenia nástroja sú prístupné cez tlačidlo nachádzajúce sa v pravom hornom rohu s názvom užívateľského účtu. [7]



Obrázok 3.2: *Prostredie nástroja Nessus*

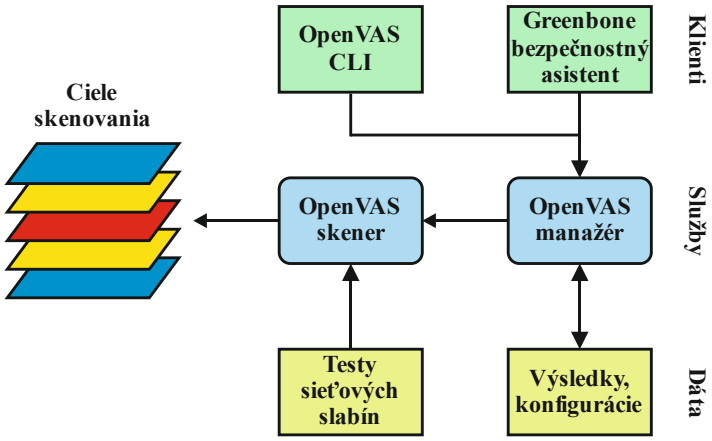
3.3 OpenVAS

Open Vulnerability Assessment System tvorí niekoľko služieb a nástrojov (obr. 3.3). Jadrom OpenVAS architektúry je OpenVAS skener zabezpečený pomocou SSL. Skener čo najefektívnejšie vykonáva testy sieťových slabín (NVT) denne aktualizovaných prostredníctvom OpenVAS NVT väzby alebo komerčných služieb.

OpenVAS manažér je centrálna služba, ktorej úlohou je pretvoriť prosté testovanie slabín do plnohodnotného manažovateľného riešenia. Manažér riadi skener cez OpenVAS Management Protocol. Všetka logika je uložená v manažérovi, takže je možná realizácia rôznych klientov s konzistentnými výsledkami. Manažér taktiež riadi SQL databázu, v ktorej sú uložené všetky konfigurácie a výsledky skenovaní a ovláda správu užívateľov, vrátane prístupových práv. [10]

K dispozícii sú dvaja rôzny klienti:

- Greenbone bezpečnostný asistent je webová služba ponúkajúca užívateľské rozhranie cez webový prehliadač.
- OpenVAS CLI je nástroj pre príkazový riadok riadiaci OpenVAS manažéra cez vytvorené procesy.



Obrázok 3.3: *Architektúra OpenVAS*

3.3.1 Prostredie OpenVAS

Nakoľko počítač, z ktorého bolo vykonávané penetračné testovanie, mal operačný systém Kali Linux, ktorý v základe obsahuje predinštalovaný OpenVAS, nebola inštalácia nutná. Na stránkach výrobcu však možno nájsť detailný návod na inštaláciu pre množstvo rôznych Linuxových distribúcii ako napr. Ubuntu, Debian, openSUSE, atď. Pred prvým spustením je vhodné spustiť najskôr úvodné nastavenie, pretože tým sa aktualizuje nástroj a všetky jeho súčasti na najnovšiu verziu. Následne potom počítač spustí alebo reštartuje OpenVAS manažéra, OpenVAS skener a Greenbone bezpečnostného asistenta a úplne na záver vytvorí účet „admin“ s náhodne vygenerovaným heslom (obr. 3.4).

```
[i] Skipping /var/lib/openvas/cert-data/dfn-cert-2014.xml, file is older than la
st revision
[i] Skipping /var/lib/openvas/cert-data/dfn-cert-2015.xml, file is older than la
st revision
[i] No DFN-CERT advisories updated and CERT DB newer than SCAP DB. Skipping CVSS
 recalculation.
Stopping OpenVAS Manager: openvasmd.
Stopping OpenVAS Scanner: openvassd.
Starting OpenVAS Scanner: openvassd.
Starting OpenVAS Manager: openvasmd.
Restarting Greenbone Security Assistant: gsad.
User created with password 'f2db23de-466a-4bf8-99ed-8e73b91ecb24'.
```

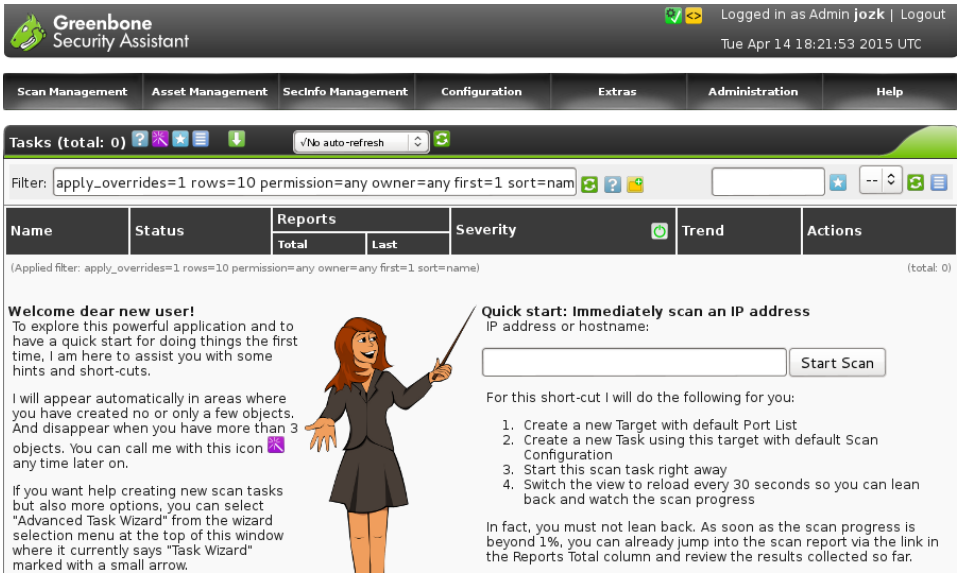
Obrázok 3.4: *Spustenie úvodného nastavenia v OpenVAS*

Aplikáciu je možné ovládať buď príkazmi zadávanými do terminálu alebo cez grafické rozhranie zvané Greenbone bezpečnostný asistent dostupné vo webovom prehliadači na adrese: „https://localhost:9392“. Obe možnosti ponúkajú rovnakú variabilitu nastavenia programu, avšak ja som si zvolil grafické prostredie, pretože tak mi ovládanie programu prišlo jednoduchšie a oveľa intuitívnejšie.

Po prihlásení do aplikácie sa zobrazí úvodná obrazovka (obr. 3.5). Tú tvorí 7 sekcií:

- Scan Management,
- Asset Management,
- SecInfo Management,
- Configuration,
- Extras,
- Administration,
- Help.

V Scan Management je možné pridávať, upravovať alebo odstraňovať úlohy a kontrolovať výsledky skenovaní. Asset Management obsahuje správu hostiteľských zariadení. V SecInfo Management sú správy o najnovšie vydaných pluginoch a iné bezpečnostné informácie. Configuration je najdôležitejšou sekciou, pretože v tejto sekcii sa nachádzajú ciele skenovaní, konfiguračné súbory s možnosťou nastavenia vlastného profilu skenovania, zoznamy skenovaných portov, filtre, povolenia, atď. Sekcia Extras informuje o dostupných výpočtových zdrojoch a všeobecných nastaveniach profilu. Aktualizácie certifikátov, sieťových testoch slabín a správu užívateľov zastrešuje sekcia Administration. [10]



Obrázok 3.5: *Prostredie nástroja OpenVAS*

3.4 Kali Linux

Kali Linux je voľne dostupná distribúcia zameraná na zjednodušenie penetračných testov. Jeho predchodcom je BackTrack, ktorý bol založený na Ubuntu. Na rozdiel od toho Kali vychádza z Debianu. Kali má vylepšené softwarové repozitáre, ktoré sú synchronizované s repozitármi Debianu, čím sa uľahčila aplikácia aktualizácii aj inštalácia nových súčastí.

Nakoľko obsahuje len nevyhnutné súčasti a balíčky, je možné prispôsobiť si Kali Linux podľa vlastných predstáv. Podporuje prostredia upravujúce plochu akými sú napr. Gnome, KDE, LXDE a iné. Vývoj Kali Linuxu bol financovaný spoločnosťou Offensive security. Offensive security je poradenská spoločnosť, zaoberajúca sa penetračným testovaním a bezpečnostnými školeniami, ktorá stála aj pri vzniku BackTrack. Preslávili sa najmä veľmi populárnym Peneterčným testovaním s BackTrack. [1]

Na rozdiel od Nessusu aj OpenVAS je Kali Linux operačným systémom a nie nástrojom k bezpečnostnému auditu. Je to upravená verzia Debianu s množstvom predinštalovaných nástrojov na penetračné testovanie. Medzi nimi je možné nájsť aj niekoľko vhodných aj k alebo zameraných výhradne na penetračné testovanie prostredia protokolu SIP. Na základe doporučení z knihy Kali Linux Cookbook som zvolil tieto nástroje:

3.4.1 Ettercap

Je všestranným nástrojom manipulujúcim so sieťou. Jeho najdôležitejšou schopnosťou je možnosť vykonávať útoky typu MITM v prepínaných sieťach. Od momentu, kedy sa dostane do pozície medzi dve komunikujúce strany, zachytáva a prehľadáva všetku premávku. Medzi ďalšie funkcie patrí:

- vkladanie znakov – vkladá ľubovoľné znaky do aktívneho obojsmerného spojenia, napodobňuje príkazy odoslané klientom alebo odpovede od serveru,
- filtrovanie paketov – automaticky filtruje užitočnú záťaž v TCP a UDP paketoch, vyhľadáva zvolené reťazce a nahrádza ich vlastnými,
- automatické zbieranie hesiel známych protokolov
- MITM útok aj na Point-to-Point Tunneling Protocol (PPTP),
- zobraziť a ukončiť zvolené pripojenie. [9]

3.4.2 Inviteflood

Je nástrojom k vykonávaniu záplavových útokov SIP/SDP správami cez UDP/IP protokol. [1]

3.4.3 Nmap

Nmap je voľne dostupný nástroj slúžiaci k prehľadávaniu siete a bezpečnostnému auditu. Nmap funguje na väčšine operačných systémov od Windowsu, cez Mac až po Linux a má grafickú nadstavbu zvanú Zenmap. [4]

3.4.4 SIPp

Je testovacím nástrojom a generátorom premávky protokolu SIP. Obsahuje niekoľko základných scenárov užívateľských agentov SipStone a vytvára a ukončuje niekoľko hovorov metódami BYE a INVITE. Všetky štatistiky prebiehajúcich testov sú dynamicky zobrazované. Môže byť použitý pri testovaní rôzneho reálneho SIP vybavenia ako napr. SIP proxy, B2BUA, SIP PBX, atď. Taktiež dokáže emulovať tisícky agentov volajúcich na zvolený SIP systém. [1]

3.4.5 **SIPVicious**

Je sada nástrojov, ktorú je možné použiť pri audite VoIP systémov založených na protokole SIP. SIPVicious pracuje na akomkoľvek systéme s podporou python verzie 2.6, prípadne novšej. Nemá grafickú nadstavbu, ovláda sa pomocou terminálu/príkazového riadku. Pozostáva zo 4 nástrojov:

- Svmap je SIP skener. Zobrazí zoznam nájdených SIP zariadení v danom adresnom rozsahu.
- Svwar vyhľadáva aktívne pravidlá (extensions) na pobočkovej ústredni.
- Svcrack prekonáva heslá na pobočkovej ústredni. Pri zadávaní príkazu je nutné stanoviť aj užívateľské meno alebo číslo, závisí od konfigurácie.
- Svreport riadi dialógy a ukladá hlásenia z testov do rôznych formátov. [1]

3.4.6 **Wireshark**

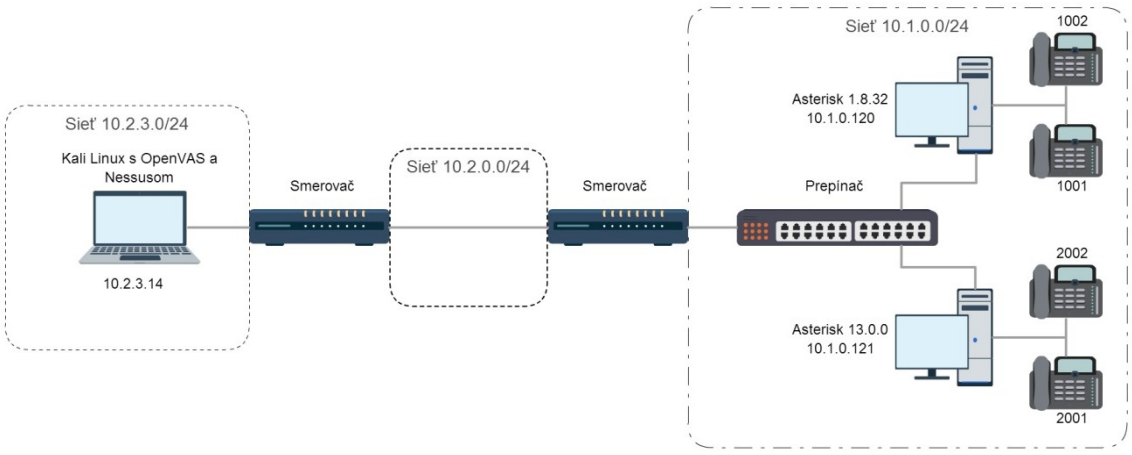
Wireshark je jedným z najpopulárnejších nástrojov sieťovej analýzy. Pri spustení na zariadení s prístupom do drôtovej alebo bezdrôtovej siete, Wireshark zachytáva a dekóduje sieťové rámce, čo z neho činí ideálny nástroj optimalizačnú, bezpečnostnú a aplikačnú analýzu. Zachytenú premávku môže uložiť do množstva rôznych formátov súborov. Proces dekódovania používa Wireshark tzv. „patológov“, ktorí identifikujú a zobrazia množstvo polí a hodnôt zo sieťových rámcov. [6]

4 Testovanie a vyhodnotenie penetračných testov

Nessus a OpenVAS sú nástroje určené k vyhľadávaniu bezpečnostných rizík pomocou nastavených profilov. Umožňujú tento proces automatizovať a výsledok vo zvolenom formáte odosielať formou hlásenia do emailovej schránky. Kali Linux je veľmi odlišný. Je to operačný systém upravený pre potreby penetračného testovania v rôznych sieťových prostrediach. Penetračné testovanie s Kali Linux vyžaduje úplne iný prístup, v ktorom nevyhľadávate bezpečnostné riziko, ale tým rizikom sa sami stávate.

4.1 Topológia a konfigurácia testovanej siete

Aby bolo možné detailnejšie otestovať a zanalyzovať schopnosti zvolených nástrojov na penetračné testovanie, bolo tomu nutné prispôsobiť aj testovanú sieť. Po odbornej konzultácii som sa rozhodol testovať na dvoch odlišných verziách Asterisku. Najstaršej, ale v súčasnosti ešte stále podporovanej verzii 1.8.X a na najnovšej verzii 13.X. Cieľom bolo zistiť či, a ak áno, tak aký veľký je rozdiel v bezpečnosti medzi týmito verziami Asteriku. Nakoľko som nedisponoval potrebným hardwarom, boli mi v školskej sieti vytvorené dva virtuálne počítače s IP adresami 10.1.0.120 a 10.1.0.121. Sieť 10.1.0.0/24 nie je dostupná z internetu, takže som sa cez VPN tunel pripájal do siete 10.2.3.0/24, z ktorej som mal prístup k obom virtuálnym strojom. Logická topológia testovanej siete mala podobu:



Obrázok 4.1: *Topológia testovanej siete*

Na oboch virtuálnych počítačoch bol nainštalovaný operačný systém Ubuntu Server vo verzii 14.04. Na počítač s IP adresou 10.1.0.120 som skompiloval a nainštaloval Asterisk verzie 1.8.32. Najnovší Asterisk verzie 13.0.0 bol po kompilácii nainštalovaný na druhý počítač, s IP adresou 10.1.0.121. Na každom z Asteriskov som vytvoril dva účty, konkrétne v prípade Asterisku 1.8.32 to boli účty s názvom 1001 a 1002, a v prípade Asterisku 13.0.0 účty a názvom 2001 a 2002. Medzi Asteriskami som vytvoril SIP trunk, takže si všetky telefóny mohli volať navzájom. V prílohe sú uvedené všetky konfiguračné súbory z oboch Asteriskov. Na svoj notebook, z ktorého som spúšťal testy som nainštaloval Kali Linux, ktorý obsahoval OpenVAS vo verzii 4.0.2 (verzia OpenVAS skeneru) a doinštaloval som Nessus verzie 6.3.5.

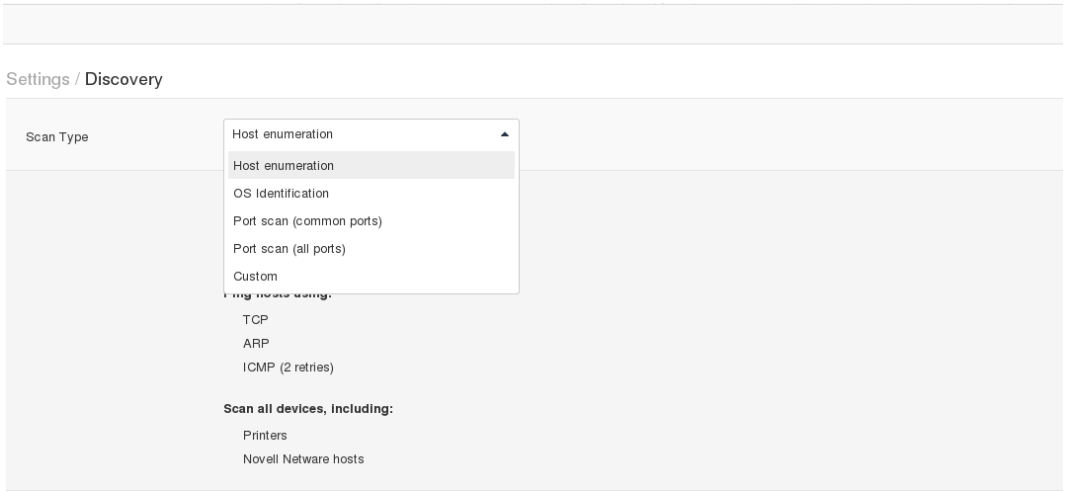
4.2 Skenovanie testovanej topológie

Skenovanie siete je prvým krokom útočníka, ktorému sa podarilo získať prístup do siete. Cieľom je odhaliť všetky aktívne zariadenia, určiť ich typ, aký majú druh operačného systému či verzie firmwaru, otvorené porty a na nich počúvajúce služby, atď.

4.2.1 Skenovanie testovanej topológie s Nessusom

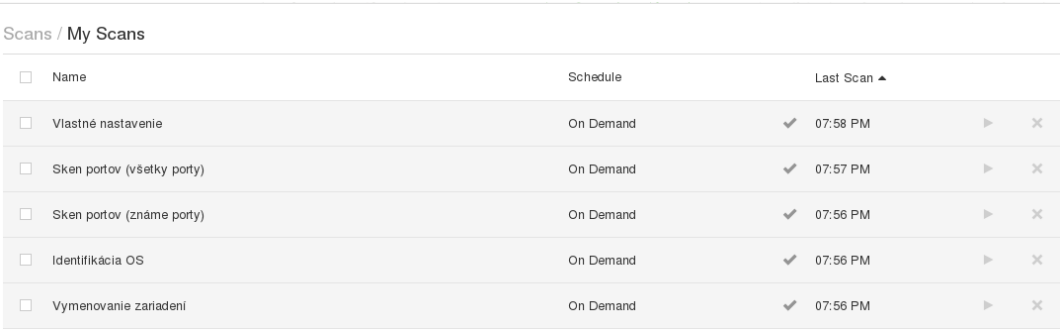
Pri skenovaní siete sa ponúka možnosť využiť prednastavený profil s názvom Host Discovery. Po nastavení názvu skenovania a stanovení IP adries, ktoré majú byť prehľadávané je možné dostať sa do časti Discovery, kde sa dá spresniť o aký typ skenovania máme záujem. Ako je možné vidieť aj na obr. 4.2, Nessus ponúka na výber 5 podprofilov určených na:

- vyhľadávanie zariadení,
- identifikáciu OS,
- skenovanie portov (známe porty),
- skenovanie portov (všetky porty),
- a profil, ktorý je možné si upraviť podľa želania.



Obrázok 4.2: *Prednastavené profily na skenovanie siete v Nessus*

Po zvolení jednej z možností sa zobrazí konfigurácia, ktorú nie je možné s výnimkou posledného podprofilu nijako upravovať. Nastavil som všetkých 5 podprofilov na testovanie zvolenej topológie (obr. 4.3). Avšak ani jeden z prednastavených podprofilov nedokázal zistiť o cieľoch uspokojivé množstvo informácii. Podprofil vyhľadávajúci aktívne zariadenia naozaj len vyhľadal aktívne zariadenia, podprofil identifikujúci operačný systém vyhľadával aktívne zariadenia a identifikoval OS, zvyšné tri podprofily kontrolovali otvorené porty TCP a UDP protokolov v stanovenom rozsahu.



Obrázok 4.3: *Všetky prednastavené podprofily na skenovanie siete testujúce zvolenú topológiu*

Nakonfiguroval som si vlastný profil skenujúci sieť. Po kliknutí na tlačidlo New Scan som zvolil možnosť Advanced Scan, ktorá ponúka najširšie možnosti konfigurácie profilu. Profil som pomenoval a za cieľ som zvolil oba počítače s Asteriskom. Prešiel som do sekcie Discovery, kde som povolil ping pomocou ARP, ICMP, UDP aj TCP protokolu, skenovanie portov UDP protokolu, v prípade TCP a SYN skenerov obísť softwarovo automatickú detekciu firewallu a zmenil vyhľadávanie SSL nielen na známych, ale na všetkých portoch. V sekcii Assessment som povolil vykonávať dôkladné testy a skenovať webové aplikácie. V sekcii Report bola zvolená možnosť zobrazovať toľko informácií, koľko je len možné a taktiež skontrolovať chýbajúce aktualizácie. V poslednej sekcii Advanced som umožnil spomaliť rýchlosť skenovania v prípade detekcie preťaženia siete. Nakoľko pri skenovaní siete nie je nutné mať k dispozícii všetky rodiny pluginov, zvolil som len General, Service detection, Port scanners a SNMP. Takto nakonfigurovaný profil dokázal zmapovať celú cestu paketu až k cieľu, zistiť verziu SSH a Asteriskov, identifikovať operačný systém, detegovať skinny server (súčasť Asterisku), nájsť otvorené TCP porty 22 (SSH) a 2000 (SCCP) a UDP port 4569 (IAX) a určiť, že skenované zariadenie môže slúžiť k rôznym účelom, nie len konkrétnej aplikácii (obr. 4.4).

Severity ▲	Plugin Name	Plugin Family	Count
INFO	Nessus SYN scanner	Port scanners	2
INFO	Nessus TCP scanner	Port scanners	2
INFO	Common Platform Enumeration (CPE)	General	1
INFO	Device Type	General	1
INFO	ICMP Timestamp Request Remote Date Disclosure	General	1
INFO	Inter-Asterisk eXchange Protocol Detection	Service detection	1
INFO	IP Protocols Scan	General	1
INFO	Nessus Scan Information	Settings	1
INFO	OS Identification	General	1
INFO	Service Detection	Service detection	1
INFO	Session Initiation Protocol Detection	Service detection	1

Obrázok 4.4: *Výsledok skenovania siete pomocou upraveného profilu*

4.2.2 Skenovanie testovacej topológie s OpenVAS

Pred sieťového skenovania je v OpenVAS nutné najskôr vytvoriť si ciele skenovania. Existujú dva možné spôsoby ako pridať cieľ skenovania. Prvým z nich je v úvodnom okne zadať IP adresu cieľového zariadenia do plochy nachádzajúcej sa napravo od obrázky slečny a potom kliknúť na tlačidlo Start Scan. Tým nielen spustíme rýchle preskenovanie zariadenia, ale zároveň ho aj pridáme medzi potenciálne ciele pri vytváraní skenovacích úloh. Toto riešenie však nie je úplne ideálne, pretože takto vytvorený ciel skenovania má prednastavený zoznam prehľadávaných portov a aj metód, pomocou ktorých sa určuje, či je zariadenie aktívne a aké služby na ňom pracujú. V prípade, že máme záujem o čo najdetailnejšie testy je k dispozícii druhá možnosť, a to v sekcii Configuration časti Targets si manuálne nakonfigurovať ciele skenovania. V tomto prípade je možné vyberať si medzi niekoľkými zoznamami portov a aj metódami, pomocou ktorých sa určuje aktívne zariadenie.

Tabulka 1.4: *Zoznamy portov používané v OpenVAS*

Názov	TCP	UDP	Celkovo
Všetky IANA TCP porty ku 10.2.2012	5625	0	5625
Všetky IANA TCP a UDP porty ku 10.2.2012	5625	5363	10988
Všetky privilegované TCP porty	1023	0	1023
Všetky privilegované TCP a UDP porty	1023	1023	2046
Všetky TCP porty	65535	0	65535
Všetky TCP a top 100 UDP portov z Nmap 5.51	65535	99	65634
Všetky TCP a top 1000 UDP portov z Nmap 5.51	65535	999	66534
Top 2000 TCP a top 100 UDP portov z Nmap 5.51	1999	99	2098
Prednastavené pre OpenVAS	4481	0	4481

Pre oba počítače s Asteriskom som zvolil možnosť „Všetky TCP a top 1000 UDP portov z Nmap 5.51“ a povolil overovanie či sú stroje aktívne pomocou ICMP, TCP aj ARP protokolov.

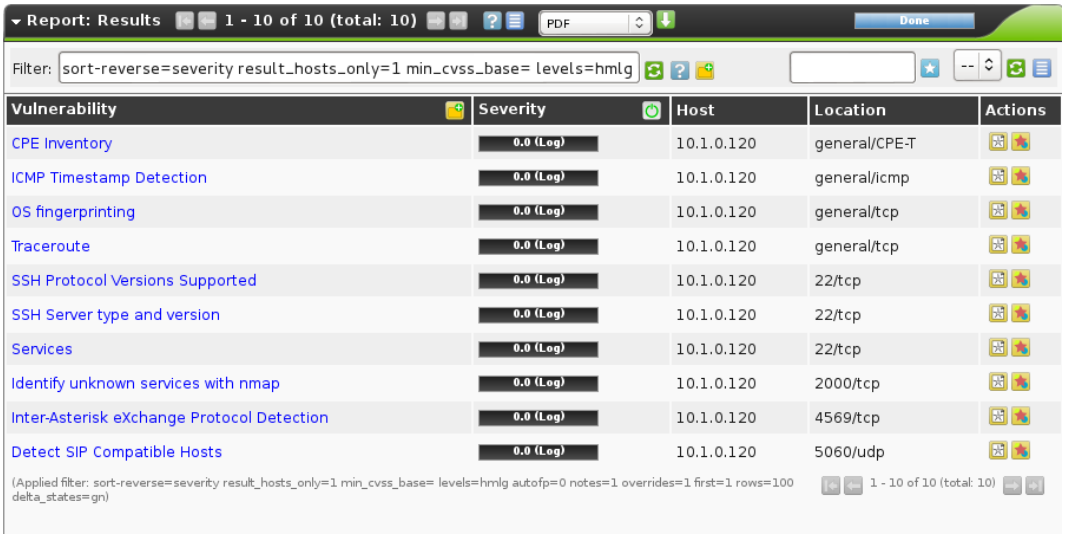
V sekcii Configuration, časti Scan Configs sa nachádzajú 3 prednastavené profily určené k skenovaniu siete a zisku čo najväčšieho množstva informácii:

- Discovery skenujúci konfiguráciu sieťových zariadení,
- Host Discovery vyhľadávajúci aktívne zariadenia,
- System Discovery skenujúci konfiguráciu sieťových systémov.

Aj v tomto prípade som nechal všetky tri prednastavené profily preskenovať testovaciu topológiu (obr. 4.5).

Obrázok 4.5: *Všetky prednastavené profily v OpenVAS na skenovanie siete testujúce zvolenú topológiu*

Tak ako je možné vidieť aj na obrázku 4.6, profil Discovery identifikoval operačný systém, zmapoval cestu paketu k cieľu, detegovaloba Asterisky, SSH a aj ich verzie a zistil, že stroje počúvajú na TCP portoch s číslom 22 (SSH) a 2000 (SCCP) a UDP portoch 4596 (IAX) a 5060 (SIP). Profil Host Discovery len vyhľadal aktívne zariadenia a System Discovery sa neúspešne pokúsil o identifikáciu operačného systému, dokázal však nájsť na porte číslo 22 protokolu TCP počúvajúcu službu SSH a určiť jej verziu.

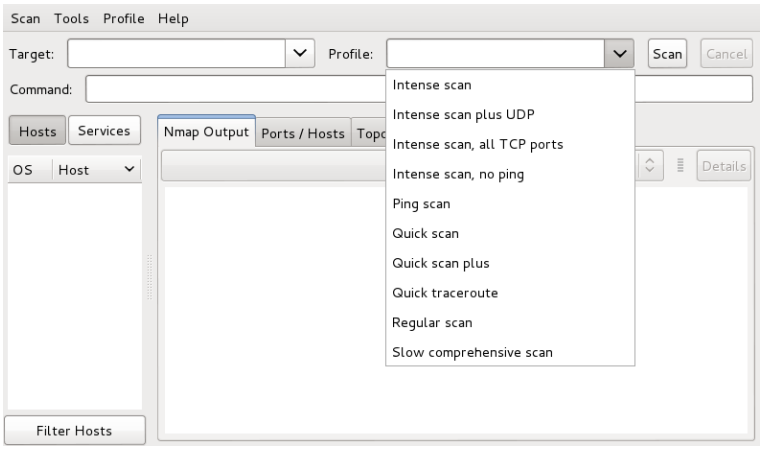


Obrázok 4.6: *Výsledok skenovania siete profilom Discovery*

Na záver som si nakonfiguroval v sekcii Configuration časti Scan Configs vlastný profil na skenovanie siete. Z rodín pluginov som povolil General, Nmap NSE, Nmap NSE net, Policy, Port scanners, Product detection, SNMP, Service detection a Useless services a v Scanner Preferencies sieťový sken. Takto nastavený profil nedokázal zistiť o testovacej topológii viac informácii, než prednastavený profil Discovery.

4.2.3 Skenovanie testovacej topológie s Kali Linux

V Kali Linux sa na skenovanie siete používa nástroj Nmap, ktorý ponúka grafickú nadstavbu zvanú Zenmap. Tá výrazne zjednodušuje prácu s nástrojom, nakoľko nie je nutné poznať syntax príkazov, ani všetky parametre zadávané do terminálu. Nmap je nástrojom určeným výhradne k skenovaniu siete, takže grafická nadstavba obsahuje hneď niekoľko prednastavených profilov umožňujúcich rôzne druhy skenovania sieťových zariadení (obr. 4.7). [4]



Obrázok 4.7: *Grafická nadstavba nástroja Nmap*

Na úvod je vhodné si preskenovať celý adresný rozsah a nájsť tak všetky aktívne zariadenia. Všetko čo musí užívateľ vykonať je do poľa Target zadať rozsah adries, ktorý chce preskenovať a v poli Profile zvoliť profil Ping scan. Zenmap sám upraví príkaz do tvaru:

```
nmap -sn rozsah_IP_adries
```

Pri skenovaní testovanej topológie pomocou profilu Ping scan sa na výstupe programu objavilo:

```
Starting Nmap 6.47 ( http://nmap.org ) at 2015-03-02 10:12 CEST
Nmap scan report for 10.1.0.120
Host is up (0.0040s latency).
Nmap scan report for 10.1.0.121
Host is up (0.0023s latency).
Nmap done: 2 IP addresses (2 hosts up) scanned in 0.13 seconds
```

Takýto výpis však neposkytuje žiadne informácie okrem jedinej, a to, že obe zariadenia sú aktívne. Zenmap podobne ako Nessus alebo OpenVAS umožňuje nakonfigurovať si vlastný profil skenovania siete. Po zadaní klávesovej skratky Crtl a E sa zobrazí dialógové okno, v ktorom je možné profil detailne upravovať. Opäť som mal záujem dostať na výstupe z nástroja o zariadeniach toľko informácii, koľko je nástroj schopný zistiť a tak som v časti Scan zadal obe IP adresy testovaných strojov, zvolil testovanie otvorených portov protokolu TCP odosielaním TCP SYN paketov a skenovanie portov UDP protokolu, povolil všetky pokročilé agresívne nastavenia, detekciu verzií spustených služieb a operačného systému. Výsledný príkaz mal tvar:

```
nmap -sS -sU -sV -T4 -O -A 10.1.0.120-121
```

Výstup z programu bol príliš dlhý na to, aby som ho uviedol celý, spomeniem tak aspoň najdôležitejšie časti, ktoré boli zistené:

```
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         (protocol 2.0)
2000/tcp open  cisco-sccp?
5060/udp open  sip         Asterisk PBX 1.8.32.2
==============NEXT SERVICE FINGERPRINT (SUBMIT
INDIVIDUALLY)==============
SF-Port22-TCP:V=6.47%I=7%D=4/28%Time=553F5D78%P=x86_64-unknown-
linux-gnu%r
SF:(NULL,27,"SSH-2\.0-OpenSSH_6\.6p1\x20Ubuntu-2ubuntu1\r\n");
TRACEROUTE (using port 111/tcp)
HOP RTT      ADDRESS
1   2.05 ms  10.2.3.1
2   11.37 ms 10.2.0.1
3   11.45 ms 10.1.0.120
```

Z výpisu je možné zistiť, že Nmap s takto zadanými parametrami príkazu pri skenovaní testovanej topológie našiel dva otvorené porty TCP protokolu s číslom 22 a 2000 a jeden UDP port 5060, identifikoval služby počúvajúce na týchto portoch, detegoval SSH, Asterisk aj ich verzie, zmapoval cestu paketu k cieľu a určil operačný systém.

Tým sa však možnosti skenovania siete a získavania čo možno najväčšieho množstva informácii v prípade Kali Linux nekončia. Kali Linux totiž obsahuje balík SIPVicious, ktorý slúži k auditu VoIP systémov založených na protokole SIP. SIPVicious nemá grafickú nadstavbu, takže pri jeho ovládaní je užívateľ odkázaný výhradne na použitie terminálu. Skladá sa zo 4 nástrojov: svmap, svwar, svcrack a svreport.

Svmap je SIP skener. Zobrazí zoznam nájdených SIP zariadení v danom adresnom rozsahu. Nástroj sa spúšťa zadaním príkazu:

```
svmap rozsah_IP_adries
```

Svwar vyhľadáva aktívne pravidlá (extensions) na pobočkovej ústredni. Príkaz by mal (optimálne) obsahovať rozsah vyhľadávaných pravidiel a pri zvolení metódy INVITE, ktorou sa vyhľadávajú aktívne pravidlá je treba vziať v úvahu, že môže spôsobiť zvonenie telefónu.

```
svwar –e rozsah_pravidiel IP_adresa --method=zvolená_metóda
```

Svcrack prekonáva heslá na pobočkovej ústredni. Pri zadávaní príkazu je nutné stanoviť aj užívateľské meno alebo číslo a rozsah čísiel testovaných pri prelamovaní hesla:

```
svcrack –u užívateľské_meno –r rozsah_testovaných_hesiel
IP_adresa
```

Svreport riadi dialógy a ukladá hlásenia z testov do rôznych formátov. V časti príkaz sa volí operácia. Je na výber medzi zobrazením všetkých skenovaní, ich prevedením do zvoleného formátu, odstránením skenovania alebo zobrazením štatistík.

```
svreport [príkaz] [nastavenia]
```

Na obrázku 4.8 je možné vidieť použitie nástrojov SIPVicious na testovanej topológii aj s príkladmi príkazov a výstupmi zobrazenými po ich spracovaní. Najskôr som pomocou svmap preskenoval oba počítače. Svmap dokázal nájsť a správne určiť verzie oboch Asteriskov. Následne som nástrojom svwar a metódou INVITE zistil, že na Asterisku 13.0.0 sú dve aktívne pravidlá pre telefóny s číslom 2001 a 2002. Na záver som použil svcrack na prelomenie hesla telefónu 2001.

```
root@kali:~# svmap 10.1.0.120-10.1.0.121
| SIP Device      | User Agent           | Fingerprint |
-------------------------------------------------------
| 10.1.0.120:5060 | Asterisk PBX 1.8.32.2 | disabled    |
| 10.1.0.121:5060 | Asterisk PBX 13.0.0   | disabled    |

root@kali:~# svwar -e 1000-3000 10.1.0.121 --method=INVITE
WARNING:TakeASip:using an INVITE scan on an endpoint (i.e. SIP phone) may cause
it to ring and wake up people in the middle of the night
| Extension | Authentication |
-----------------------------
| 2002      | reqauth        |
| 2001      | reqauth        |

root@kali:~# svcrack -u2001 -r1500-2500 10.1.0.121
| Extension | Password |
------------------------
| 2001      | 2001     |
```

Obrázok 4.8: *Použitie SIPVicious na testovanej topológii*

4.2.4 Vyhodnotenie a porovnanie výsledkov skenovania testovacej topológie

Cieľom skenovania siete bolo zistiť pokiaľ možno čo najviac informácii o zariadeniach nachádzajúcich sa v nej nielen po hardwarovej, ale hlavne po softwarovej stránke.

Pri dôkladnom skenovaní siete dokázal Nessus pomocou upraveného profilu zmapovať celú cestu paketu až k cieľu, zistiť verziu SSH a Asteriskov, identifikovať operačný systém, detegovať skinny server (súčasť Asterisku), nájsť otvorené TCP porty 22 (SSH) a 2000 (SCCP) a UDP port 4569 (IAX) a určiť, že skenované zariadenie môže slúžiť k rôznym účelom, nie len konkrétnej aplikácii.

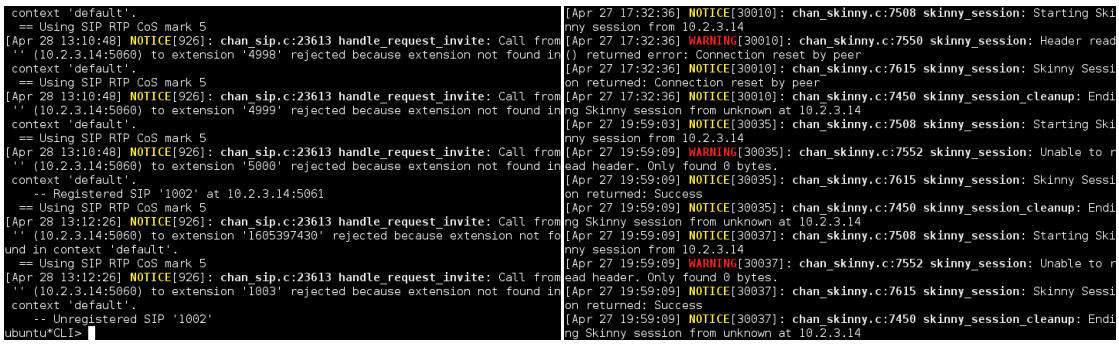
Ukázalo sa, že na dôkladné skenovanie siete nebolo v prípade testovania využívajúceho OpenVAS nutné nastavovať vlastný profil, rovnako dobre poslúžil aj prednastavený profil Discovery. Profil Discovery identifikoval operačný systém, zmapoval cestu paketu k cieľu, detegoval oba Asterisky, SSH a aj ich verzie a zistil, že stroje počúvajú na TCP portoch s číslom 22 (SSH) a 2000 (SCCP) a UDP portoch 4596 (IAX) a 5060 (SIP).

V Kali Linux sa po zadaní správnych parametrov nástroju Nmap nájsť dva otvorené porty TCP protokolu s číslom 22 a 2000 a jeden UDP port 5060, identifikovať služby

počúvajúce na týchto portoch, detegovať SSH, Asterisk aj ich verzie, zmapovať cestu paketu k cieľu a určiť operačný systém na oboch strojoch. Balík nástrojov SIPVicious k tomu pridáva ešte vyhľadanie aktívnych pravidiel a možnosť prelomenia hesiel.

Na rozdiel od Nessusu a OpenVAS Kali Linux nezistil na UDP porte 4596 IAX protokol, Nessus ako jediný určoval typ zariadenia a Kali Linux ako jediný vyhľadával aktívne pravidlá a hrubou silou na nich zisťuje heslá. Vo výsledku to znamená, že po preskenovaní testovacej topológie Nessus a Kali Linux poskytli 9 užitočných informácii, OpenVAS o jednu menej.

Tak ako bolo poznamenané už v 2. kapitole, skenovanie siete je veľmi jednoducho odhaliteľné. V prípade skenovania siete ktorýmkoľvek zo spomínaných nástrojov, som mohol po prihlásení sa na Asterisky v debug móde pozorovať množstvo premávky, pochádzajúcej práve od skenerov (obr. 4.9).



Obrázok 4.9: *Výpis z Asteriskov 1.8.32 (vľavo) a 13.0.0 (vpravo) počas skenovania siete nástrojmi svwar (vľavo) a Nessus (vpravo)*

4.3 Penetračné testovanie na odolnosť voči DoS

Útoky tohto typu bránia legitímnym užívateľom používať sieť. Medzi tri najčastejšie útoky patrí preťaženie služby, záplava správami a narušenie signalizácie. [5]

4.3.1 Testovanie topológie s Nessusom na odolnosť voči DoS

Nessus neobsahuje žiaden prednastavený profil určený k penetračnému testovaniu siete na odolnosť voči DoS a DDoS útokom, ale jedna z rodín pluginov sa zaoberá špeciálne týmto typom útoku. Jedinou možnosťou je teda vytvoriť si vlastný profil. Pomenoval som profil a ako cieľ testu som zvolil oba virtuálne počítače. V sekcii Discovery som zakázal úplne všetky možnosti, pretože o skenovanie siete som nemal v tomto prípade absolútne žiadny záujem. V sekcii Assessment som upravil presnosť testovania na zobrazovanie varovaní o potenciálnych chybách a povolil možnosť vykonávať hĺbkové testy. V sekcii Report som nastavil nástroj tak, aby zobrazoval čo možno najviac informácii je možné a aby vyhľadával chýbajúce aktualizácie. V sekcii Advanced som ponechal povolené bezpečnostné kontroly a pridal som ešte možnosť spomaliť skenovanie v prípade zahltenia siete a detekciu preťaženia Linuxového jadra. Z rodín pluginov boli počas testu povolené len 3, a to Denial of Service, Misc., pretože táto rodina pluginov obsahuje množstvo testov širokej škály softwaru a Ubuntu Local Security Checks,

nakoľko oba Asterisky sú spustené na Ubuntu server 14.04 a DoS útok nemusí mať priamo za cieľ len Asterisk, ale aj napr. operačný systém, na ktorom je služba spustená. [11]

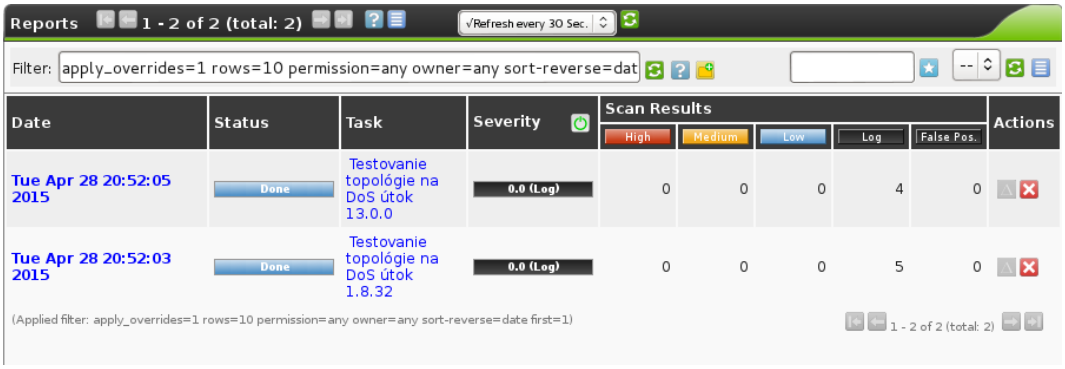
S takto nakonfigurovaným profilom Nessus odhalil na testovacej topológii hneď niekoľko bezpečnostných slabín (obr. 4.10) s rôznou úrovňou ohrozenia. Len niekoľko z nich sa však týkalo útokov typu DoS, k tým zvyšným sa ešte vrátim. Plugin s ID 79439 objavil chybu v ovládači chan_pjsip, ktorú by mohol vzdialený útočník využiť k zastaveniu aplikácie a chybu v module res_pjsip_refer, ktorými je možné pri odosielaní špeciálne sfalšovaných žiadostí zastaviť aplikáciu. Podľa pluginu s ID 81205 obsahuje Asterisk 13.0.0 chybu, kvôli ktorej zlyhá pri uvoľňovaní RTP portov pridelených spojeniu s autentizovaným koncovým bodom podporujúcim len kódeky, ktoré nie sú povolené v Asterisku. Útočník by túto chybu mohol zneužiť k vyčerpaniu dostupných portov, čo by znamenalo zastavenie služby. Plugin s ID 80036 popisuje dvojitú chybu v module res_http_websocket spracovávajúcim užitočnú záťaž nulovej dĺžky, ktorá umožňuje DoS útok. Všetky tieto chyby a týkali výhradne Asterisku verzie 13.0.0, na Asterisku 1.8.32 nebola detegovaná žiadna chyba, ktorá by mohla viesť k útoku typu DoS.

Severity ▲	Plugin Name	Plugin Family	Count
HIGH	Asterisk ConfBridge 'dialplan' Privilege Escalation (AST-2014-017)	Misc.	1
MEDIUM	Asterisk 'res_http_websocket' Double-Free DoS (AST-2014-019)	Misc.	1
MEDIUM	Asterisk chan_pjsip Incompatible Codecs DoS (AST-2015-001)	Misc.	1
MEDIUM	Asterisk libcURL HTTP Request Injection (AST-2015-002)	Misc.	1
MEDIUM	Asterisk Multiple Vulnerabilities (AST-2014-012 / AST-2014-018)	Misc.	1
MEDIUM	Asterisk PJSIP Multiple Vulnerabilities (AST-2014-013 / AST-2014-015 / AST-20...	Misc.	1
MEDIUM	Asterisk TLS Certificate Common Name NULL Byte Vulnerability (AST-2015-003)	Misc.	1
MEDIUM	OpenSSH SSHFP Record Verification Weakness	Misc.	1
LOW	SSH Server CBC Mode Ciphers Enabled	Misc.	1
LOW	SSH Weak MAC Algorithms Enabled	Misc.	1

Obrázok 4.10: *Bezpečnostné ohrozenia odhalené Nessusom testujúcim podľa profilu na vyhľadávanie slabín vedúcim k DoS na Asterisku verzie 13.0.0*

4.3.2 Testovanie topológie s OpenVAS na odolnosť voči DoS

Rovnako ako v prípade Nessusu, ani OpenVAS nemá žiaden prednastavený profil, ktorý by bol učený na testovanie odolnosti siete voči DoS útokom, ale taktiež obsahuje rodinu pluginov zameranú na vyhľadávanie bezpečnostných slabín vedúcich k DoS útoku. Pri vytváraní profilu na testovanie odolnosti voči DoS útokom som zvolil 3 rodiny pluginov: Denial od Service, General (obsahuje všeobecné testy sieťových slabín) a Ubuntu Local Security Checks. V Scanner Preferences som povolil možnosť zaznamenávať celý útok. OpenVAS s takto nakonfigurovaným profilom nenašiel žiadne bezpečnostné riziko (obr. 4.11).



Obrázok 4.11: *Výsledok testovania topológie na odolnosť voči DoS útokom nástrojom OpenVAS*

4.3.3 Testovanie topológie s Kali Linux na odolnosť voči DoS

Kali Linux nemá žiaden nástroj, pomocou ktorého by bolo možné vyhľadávať chyby v aplikáciách ústiace v riziko útoku DoS. Kali Linux k testovaniu odolnosti siete voči DoS útokom používa nástroje generujúce obrovské množstvo správ, ktorými zaplavuje ciele útoku.

Jedným z nich je Inviteflood, ktorý slúži k záplavovým útokom SIP/SDP správami cez UDP protokol. Nástroj sa spúšťa cez terminál príkazom:

```
inviteflood výstupné_sieťové_rozhranie názov_cieľového_užívateľa
cieľová_doména IP_adresa_cieľového_užívateľa počet_paketov
```

V príkaze je možné špecifikovať ešte niekoľko ďalších parametrov:

- -a pridá meno, ktoré sa zobrazí na strane obete,
- -i nastavenie zdrojovej IP adresy,
- -S nastavenie zdrojového čísla portu,
- -D nastavenie cieľového čísla portu,
- -l nastavenie textu v správach (pôvodne prázdne správy),
- -t použitie TCP protokolu miesto UDP,
- -s čas medzi odosielaním jednotlivých žiadostí (v sekundách),
- -v vypíše všetky odoslané žiadosti do terminálu.

Ako prvý som testoval Asterisk verzie 1.8.32. Inviteflood som nastavil tak, aby odoslal 1 000 000 INVITE žiadostí na účet 1001 a testoval som pripojenie (obr. 4.12). Počas trvania útoku nebolo možné vytvoriť akékoľvek spojenie, ale na spojenie vytvorené pred zahájením útoku to nemalo žiaden vplyv. Tieto isté výsledky som dostal aj po otestovaní nástroja na Asterisku verzie 13.0.0.

```
root@kali:~/Desktop# inviteflood tun0 1001 10.1.0.120 10.1.0.120 1000000

inviteflood - Version 2.0
              June 09, 2006

source IPv4 addr:port    = 10.2.3.14:9
dest   IPv4 addr:port    = 10.1.0.120:5060
targeted UA              = 1001@10.1.0.120

Flooding destination with 1000000 packets
sent: 1000000
```

Obrázok 4.12: *Inviteflood simulujúci DoS útok na Asterisk 1.8.32*

Druhým nástrojom schopným simulovať záplavový DoS útok je SIPp. SIPp je testovacím nástrojom a generátorom premávky obsahujúcim niekoľko základných scenárov užívateľských agentov. Nastavením „správnych“ parametrov je možné SIPp nakonfigurovať tak, aby generoval obrovské množstvo INVITE žiadostí a zaplavil nimi požadovaný cieľ.

Pri testovaní topológie som postupoval tak, ako v prípade nástroja Inviteflood. Spustil som nástroj príkazom:

```
sipp uac -r 10000 -rp 1s IP_adresa_Asterisku
```

Uac znamená, že bol zvolený scenár užívateľský agent klient, parameter –r stanovuje počet odoslaných INVITE žiadostí a parameter –rp za akú jednotku času sú tieto žiadosti odosielané. Na ani jednom z Asterikov sa počas toho, ako boli cieľom útoku, nedalo vytvoriť nové spojenie, avšak opäť útok nemal vplyv na už existujúce spojenia. Výhodou SIPp oproti Inviteflood je, že nie je nutné zadávať počet odosielaných paketov, takže dokáže pracovať aj dlhodobo a dynamicky zobrazuje štatistiky o odosielaných žiadostiach (obr. 4.13).

```
------------------------------ Test Terminated --------------------------------


---------------------------- Statistics Screen ------- [1-9]: Change Screen --
  Start Time              |             23:56:51:299    1430258211.299216

  Last Reset Time         |             23:56:55:322    1430258215.322121

  Current Time            |             23:56:55:629    1430258215.629709

-------------------------+---------------------------+--------------------------
  Counter Name            | Periodic value            | Cumulative value
-------------------------+---------------------------+--------------------------
  Elapsed Time            | 00:00:00:307              | 00:00:04:330
  Call Rate               | 1465.798 cps              | 8386.144 cps
-------------------------+---------------------------+--------------------------
  Incoming call created   |        0                  |        0
  OutGoing call created   |      450                  |    36312
  Total Call created      |                           |    36312
  Current Call            |    30000                  |
-------------------------+---------------------------+--------------------------
  Successful call         |        0                  |        0
  Failed call             |      450                  |     6312
-------------------------+---------------------------+--------------------------
  Response Time 1         | 00:00:00:000              | 00:00:00:000
  Call Length             | 00:00:02:116              | 00:00:00:814
------------------------------ Test Terminated --------------------------------
```

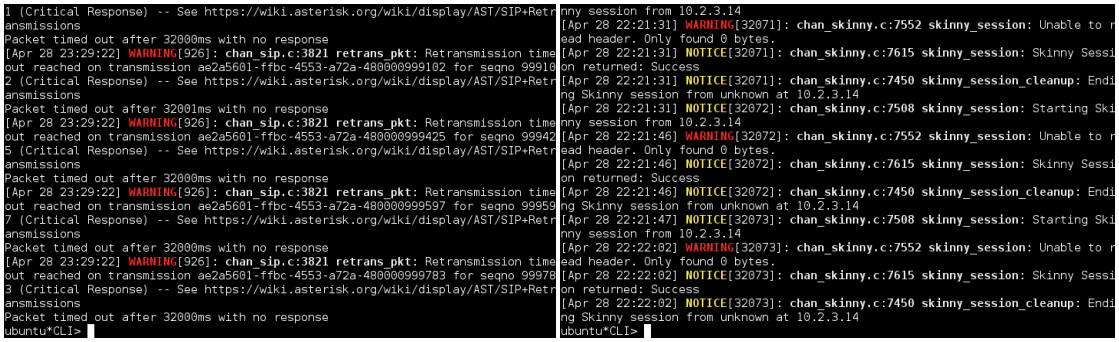
Obrázok 4.13: *Štatistiky zobrazované nástrojom SIPp*

4.3.4 Vyhodnotenie a porovnanie výsledkov testov odolnosti voči útokom typu DoS

Pri týchto testoch sa preukázali veľké rozdiely medzi jednotlivými nástrojmi a hlavne v prístupe k penetračným testom. Zatiaľ čo Nessus vyhľadával chyby, ktoré do Asteriskov neúmyselne naprogramovali tvorcovia, nástrojmi z Kali Linux som napodobňoval správanie útočníkov. Taktiež vidieť veľké rozdiely medzi Nessusom a OpenVAS, hoci by sa o týchto dvoch nástrojoch dalo povedať, že sú vzdialení príbuzní (OpenVAS vychádza z poslednej voľne dostupnej verzie Nessusu z roku 2005). 10 rokov vývoja v tomto prípade znamenal rozdiel medzi odhalením 4 chýb v kóde Asterisku 13.0.0 a neodhalením vôbec žiadnej.

Inými slovami, použitím Nessusu boli upraveným profilom detegované 4 chyby, ktoré by mohli viesť k útoku typu DoS. Všetky 4 chyby sa týkali výhradne Asterisku verzie 13.0.0. OpenVAS nenašiel v testovacej topológii žiadne bezpečnostnú slabinu, ktorá by mohla vyústiť v prerušenie alebo spomalenie fungovania Asteriskov. Kali Linux bolo možné prostredníctvom nástrojom Inviteflood aj SIPp zahltiť oba Asterisky INVITE žiadosťami do takej miery, že neboli schopné ďalej vykonávať ďalšiu činnosť. To znamená, že Nessus odhalil 4 bezpečnostné medzery, ktoré by mohli potenciálne prerásť v zastavenie služby, Kali Linux odhalil, že oba Asterisky je možné odstaviť z činnosti zaplavením INVITE žiadostí a OpenVAS nezistil žiadne bezpečnostné riziko takéhoto charakteru.

V prípade útokov záplavou INVITE žiadostí, nie je prekvapením, že sa Asteriskoch spustených v móde na odstránenie chýb zobrazilo enormné množstvo informácii, a že takýto útok by bol jednoducho odhaliteľný. Táto situácia sa však zopakovala, aj keď s oveľa menším množstvom výpisov, aj v prípade testovania topológie upravenými profilmi Nessusu aj OpenVAS. Takže, nezávisle na zvolenom nástroji, je takýto typ testovania či skenovania siete detekovateľný (obr. 4.14).



Obrázok 4.14: *Výpis z Asteriskov 1.8.32 (vľavo) a 13.0.0 (vpravo) počas testovania na odolnosť voči DoS útokom nástrojmi Inviteflood (vľavo) a OpenVAS (vpravo)*

4.4 Penetračné testovanie na detekciu možností manipulácie s registráciou

Manipulácia s registráciou má niekoľko foriem. Od ovplyvňovania registračného procesu, čo vedie k pridaniu, odstráneniu alebo ukradnutiu registrácie až po presmerovanie hovorov. Reláciu môže ovplyvniť aj vkladanie SIP správ do siete.

4.4.1 Testovanie topológie Nessusom na možnosti manipulácie s registráciou

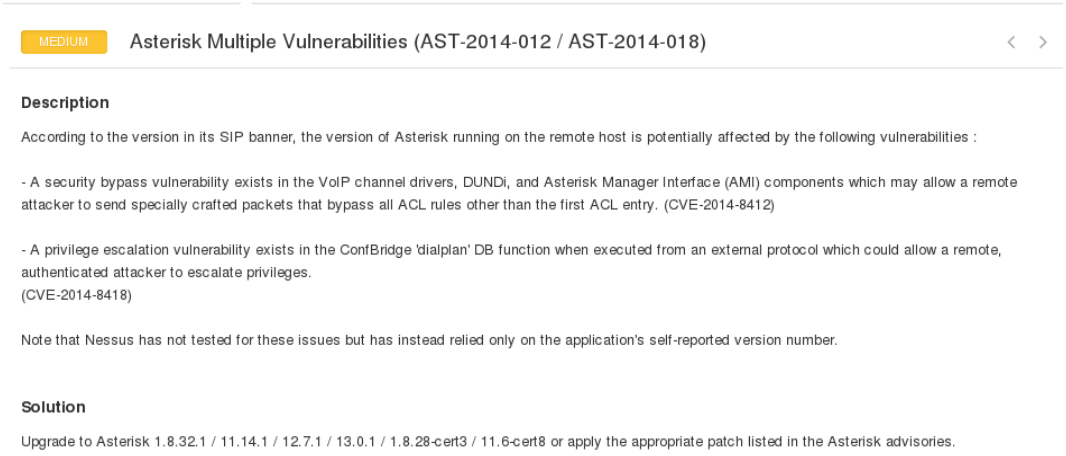
Medzi šablónami skenovaní nie je v Nessuse žiadna, určená k vyhľadávaniu bezpečnostným slabín umožňujúcich manipuláciu s registráciami. Tomuto druhu útoku nie je vyhradená ani žiadna rodina pluginov. Je však možné nájsť niekoľko rodín pluginov, ktoré je pre takéto testy možné použiť. Opäť som musel vytvoriť na mieru upravený profil skenovania. Pomenoval som profil a zvolil ciele testu. V sekcii Discovery som zakázal úplne všetky možnosti, v sekcii Assessment som povolil obísť normálnu presnosť, zobrazovať potenciálne chybové hlásenia a zakázal vyžadovať informácie o SMB doméne. V sekcii Report som vyžiadal zobrazovať maximálne množstvo informácii a vyhľadávať chýbajúce aktualizácie. V sekcii Advanced boli povolené len možnosti povolenia bezpečnostných kontrol, spomalenia testu v prípade zahltenia siete a detekcie preťaženia Linuxového jadra. Z rodín pluginov boli povolené len:

- Backdoors – detegujú trójske kone, červy a znaky napadnutého systému,
- Brute force attack – útoky hrubou silou na heslá,
- Default Unix Accounts – kontroluje prítomnosť prednastavených účtov v Unixových a linuxových systémoch,
- Gain a shell remotely – testy bezpečnostných slabín širokej škály softwaru umožňujúceho vzdialené ukončovanie kódu alebo príkazu,
- Misc. – testy veľkého množstva softwaru,
- Ubuntu Local Security Checks – bezpečnostná kontrola Ubuntu Linux systémov.

Testovaním Nessus odhalil niekoľko bezpečnostných nedostatkov, no opäť sa nie všetky týkali výhradne možnosti manipulácie s registráciou. Bolo to spôsobené tým, že som použil rodinu pluginov s názvom Misc., ktorá obsahuje množstvo bezpečnostných testov širokej škály softwaru, evidentne aj Asterisku, ale tieto testy nie sú nijako rozdelené do kategórii. Aj v tomto prípade teda uvediem len nájdené chyby, týkajúce sa možnosti manipulácie s registráciou.

Plugin s ID 79438 v Asterisku verzie 13.0.0 odhalil dve chyby, ktoré som zaradil do kategórie krádež registrácie, pretože umožňujú útočníkovi získať práva resp. prístup, aký by za bežných okolností nemal (obr. 4.15). Prvá chyba sa týka komponentov: ovládač VoIP kanálu, DUNDI a Asterisk Manager Interface. Táto chyba môže útočníkovi umožniť prekonať prístupové zoznamy (ACL, access list). Pomocou druhej chyby z funkcie ConfBridge 'dialplan' DB môže útočník získať oprávnenia. Plugin s ID 81257 našiel chybu vo funkcii 'parseurlandfillconn', ktorú by mohol útočník využiť k odosielaniu neautorizovaných HTTP

žiadostí obsahujúcich škodlivé dáta. Štvrtá a zároveň aj posledná chyba objavená pluginom s ID 79441 popisuje chybu vo funkcii ConfBridge 'dialplan' objavujúcej sa pri spúšťaní z externého protokolu. Touto chybou útočník môže získať oprávnenia k spúšťaniu ľubovoľných systémových príkazov.



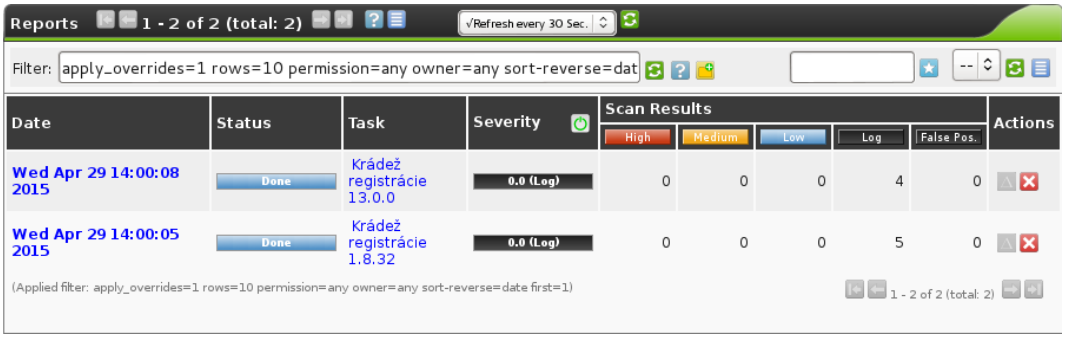
Obrázok 4.15: *Popis dvoch chýb v Asterisku verzie 13.0.0 nájdených pluginom s ID 79438*

4.4.2 Testovanie topológie s OpenVAS na možnosti manipulácie s registráciou

Tak ako v prípade Nessusu, ani OpenVAS nemá vopred pripravený profil, pomocou ktorého by OpenVAS vyhľadával bezpečnostné medzery s rizikom manipulácie s registráciou. Takže aj v tomto prípade bolo nutné vytvoriť a účelovo upraviť nový profil testovania. Zvolil som možnosť prázdny a statický, pretože v tejto konfigurácii bude obsahovať len testy nevyhnutné k určovaniu bezpečnostných slabín týkajúcich sa možností manipulácie s registráciou. Z rodín pluginov som povolil:

- Brute force attack – útoky hrubou silou na heslá,
- Defaultné účty - kontroluje prítomnosť prednastavených účtov v Unixových a linuxových systémoch,
- Gain a shell remotely – testy bezpečnostných slabín širokej škály softwaru umožňujúceho vzdialené ukončovanie kódu alebo príkazu,
- General - testy veľkého množstva softwaru,
- Malware – škodlivý software,
- Privilege escalation – získanie oprávnenia,
- Ubuntu Local Security Checks – bezpečnostná kontrola Ubuntu Linux systémov.

Spustil som testy na oba virtuálne počítače s Asteriskom, no po ich dokončení OpenVAS nezistil žiadne bezpečnostné ohrozenie, ktoré by mohlo spôsobiť manipuláciu s registráciou (obr. 4.16).



Obrázok 4.16: *Otestovanie topológie nástrojom OpenVAS s profilom na vyhľadávanie ohrozenia manipulácie s registráciou*

4.4.3 Testovanie topológie s Kali Linux na možnosti manipulácie s registráciou

Kali Linux k týmto testom opäť pristupuje úplne odlišným spôsobom než Nessus a OpenVAS. K testovaniu možností manipulácie s registráciou využíva nástroj svcrack z balíka SIPVicious. Ten slúži k prelamovaniu hesiel, zasielaním REGISTRATION žiadostí, kde postupne skúša heslá z prideleného rozsahu až do momentu, kedy uspeje. Aby som dôkladnejšie otestoval tento nástroj, zmenil som heslá na účtoch 1001 a 1002, nachádzajúcich sa na Asterisku verzie 1.8.32 z 1001 na 50694 a z 1002 na 5069.

Ako prvý som testoval účet 1002. Nástroj som spustil zadaním príkazu:

```
svcrack –u1002 –v 10.1.0.120
```

Za parameter –u sa udáva číslo, ktorého heslo chceme zistiť a parameter –v znamená, že program zobrazuje aj hlásenia. Nakoľko nástroj nedokázal zistiť heslo, musel som mu cez parameter –r zadať aj rozsah hesiel, ktoré má testovať. Príkaz mal potom syntax:

```
svcrack –u1002 –r 1-99999 –v 10.1.0.120
```

Nástroju trvalo 68 sekúnd, než dokázal zistiť správne heslo. Potom som prešiel test účtu 1001. Je logické, že v tomto prípade test zabral podstatne viac času. Na prelomenie tohto hesla potreboval svcrack 688 sekúnd (obr. 4.17).

```
root@kali:~# svcrack -u1002 -v 10.1.0.120
INFO:ASipOfRedWine:trying to get self ip .. might take a while
INFO:root:scan started at 2015-04-29 16:45:24.531053
INFO:ASipOfRedWine:no more passwords
WARNING:root:found nothing
INFO:root:Total time: 0:00:22.415419
root@kali:~# svcrack -u1002 -r 1-99999 -v 10.1.0.120
INFO:ASipOfRedWine:trying to get self ip .. might take a while
INFO:root:scan started at 2015-04-29 16:46:12.861209
INFO:ASipOfRedWine:The password for 1002 is 5069
INFO:root:we have 1 cracked users
| Extension | Password |
------------------------
| 1002      | 5069     |

INFO:root:Total time: 0:01:08.671490
root@kali:~# svcrack -u1001 -r 1-99999 -v 10.1.0.120
INFO:ASipOfRedWine:trying to get self ip .. might take a while
INFO:root:scan started at 2015-04-29 16:47:30.719061
INFO:ASipOfRedWine:The password for 1001 is 50694
INFO:root:we have 1 cracked users
| Extension | Password |
------------------------
| 1001      | 50694    |

INFO:root:Total time: 0:11:28.535742
```

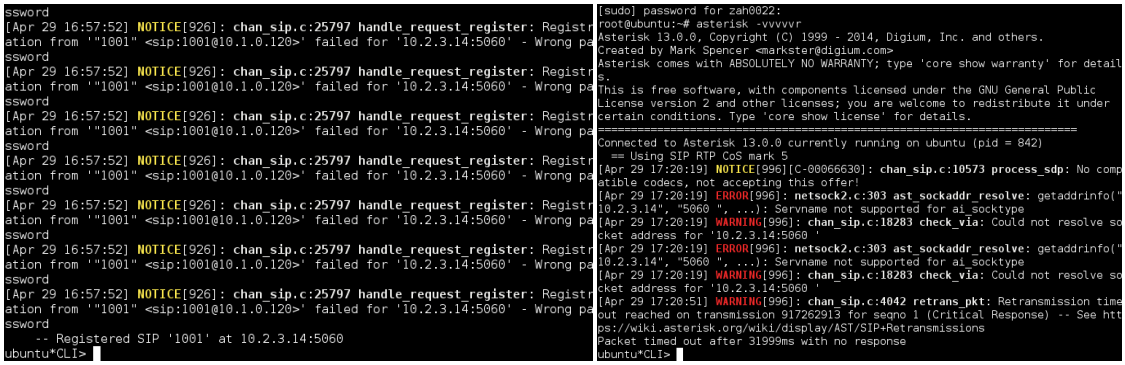
Obrázok 4.17: *Prelamovanie hesiel účtov 1001 a 1002 na Asterisku verzie 1.8.32 nástrojom svcrack*

4.4.4 Vyhodnotenie a porovnanie výsledkov testov topológie na možnosti manipulácie s registráciou

Cieľom týchto testov bolo odhaliť bezpečnostné riziká, ktoré by mohli umožniť útočníkovi pridávať, odoberať alebo ukradnúť registráciu, či presmerovať hovory.

Nessus testom odhalil, že na Asterisku vo verzii 13.0.0 sú 4 chyby, ktoré by mohli útočníkovi umožniť získať oprávnenia, k akým by inak prístup nemal. OpenVAS nedetegoval žiadne bezpečnostné ohrozenie takéhoto charakteru. Kali Linux dokáže nástrojom svcrack získavať heslá od účtov nezávisle na verzii Asterisku. Vo výsledku to znamená, že v testovacej topológii Nessus odhalil štyri chyby umožňujúce manipuláciu s registráciou, OpenVAS žiadnu a Kali Linux dve.

Za zmienku ešte určite stojí, že hoci sa všetky testy prejavili na výpisoch v Asterisku, v prípade Nessusu aj OpenVAS sa jednalo len o niekoľko krátkych záznamov (obr. 4.18), ale prekonávanie hesla svcrackom znamenalo prijatie viac ako 70 REGISTRATION žiadostí každú sekundu, čo sa prejavilo aj na spotrebovaní výpočtovej kapacity virtuálneho stroja. Niektorí sieťoví administrátori sledujú napr. pomocou protokolu SNMP štatistiky zaťaženia kriticky dôležitých zariadení a takýto útok, ktorý by sa prejavil neobvyklou aktivitou by tak jednoducho spozorovali a zastavili.



Obrázok 4.18: *Výpis z Asteriskov 1.8.32 (vľavo) a 13.0.0 (vpravo) počas testovania nástrojom svcrack (vľavo) a OpenVAS (vpravo)*

4.5 Penetračné testovanie na detekciu hrozby MITM útokov

MITM je typ útoku, kedy sa útočník dostane medzi dve komunikujúce strany s účelom odchytávať, čítať alebo dokonca pozmeňovať ich vzájomnú komunikáciu.

4.5.1 Detekcia hrozby MITM útokov v testovanej topológii Nessusom

Medzi prednastavenými profilmi skenovaní nie je ani jeden určený k vyhľadávaniu hrozby MITM útokov. To isté platí aj o rodinách pluginov, takže jedinou možnosťou bolo nakonfigurovať si vlastný profil. Zvolil som Advanced Scan, pretože iba tento profil má sprístupnené pokročilé nastavenia. Pomenoval som profil skenovania a zvolil som cieľ útoku (oba Asterisky). V sekcii Discovery som zakázal všetky možnosti, pretože o skenovanie siete som nemal záujem. V sekcii Assessment som zvýšil obvyklú presnosť na zobrazovanie potenciálnych chybových hlásení a zapol vykonávanie hĺbkových testov. V sekcii Report som povolil možnosť vyhľadávať chýbajúce aktualizácie a nastavil profil na zobrazovanie čo najväčšieho množstva informácii. V sekcii Advanced som zapol detekciu preťaženia Linuxového jadra a pridal možnosť spomaliť test v prípade zahltenia siete. Spomedzi všetkých rodín pluginov som pre tento profil skenovania vybral dve: Misc., pretože obsahuje množstvo testov na širokú paletu softwaru a Ubuntu Local Security Checks, pretože oba Asterisky sú spustené na Ubuntu server 14.04. Po skončení testu Nessus zobrazil v prípade Asterisku 1.8.32 celkovo 4 ohrozenia bezpečnosti a na Asterisku 13.0.0 to boli 3 ohrozenia bezpečnosti. Všetky sa týkali útokov typu MITM (obr. 4.19).

Severity ▲	Plugin Name	Plugin Family	Count
MEDIUM	Asterisk TLS Certificate Common Name NULL Byte Vulnerability (AST-2015-...	Misc.	1
MEDIUM	OpenSSH SSHFP Record Verification Weakness	Misc.	1
LOW	SSH Server CBC Mode Ciphers Enabled	Misc.	1
LOW	SSH Weak MAC Algorithms Enabled	Misc.	1
INFO	Nessus SYN scanner	Port scanners	2
INFO	Asterisk Detection	Misc.	1
INFO	SSH Algorithms and Languages Supported	Misc.	1

Obrázok 4.19: *Ohrozenia typu MITM nájdené Nessusom v testovanej topológii*

Plugin s ID 82901 našiel chybu v Asterisku 1.8.32. Pri registrácii SIP TLS zariadenia nedochádza k správnemu overovaniu mena serveru oproti poľu X.509 Common Name. Útočník znalý tejto slabiny by mohol prostredníctvom sfalšovaného SSL serveru zachytávať sieťovú premávku. Plugin s ID 71049 upozornil, že oba Asterisky umožňujú použitie 96 bitových MAC algoritmov, ktoré sú považované za nedostatočne bezpečné. Plugin s ID 71049 pre zmenu upozorňoval na podporu šifrovania Cipher Block Chaining, z ktorého môže útočník dešifrovať pôvodnú správu. Posledné ohrozenie bezpečnosti sa týkalo oboch virtuálnych počítačov a ich zastaralej verzie SSH, ktorá môže byť zneužitá pri podvrhnutí SSH serveru.

4.5.2 Detekcia hrozby MITM útokov v testovanej topológii OpenVAS

Keďže OpenVAS neobsahuje žiaden prednastavený profil zameraný na detegovanie hrozby útoku MITM, tak som musel nakonfigurovať profil schopný detegovať aj tento typ útoku. OpenVAS nemá medzi rodinami pluginov žiadnu, určenú k detekcii hrozby MITM útoku, preto som musel zvoliť len rodiny pluginov General, ktorá obsahuje množstvo rôznych typov testov na širokú paletu softwaru a Ubuntu Local Security Checks. Ani v tomto prípade OpenVAS nedetegoval žiadnu hrozbu, a už vôbec nie hrozbu útoku MITM (obr. 4.20).

Reports 1 - 1 of 1 (total: 1) √No auto-refresh

Filter: apply_overrides=1 rows=10 permission=any owner=any sort-reverse=dat

Date	Status	Task	Severity	Scan Results					Actions
				High	Medium	Low	Log	False Pos.	
Wed Apr 29 20:39:39 2015	Done	Test na vyhľadávanie MITM hrozieb	0.0 (Log)	0	0	0	10	0	

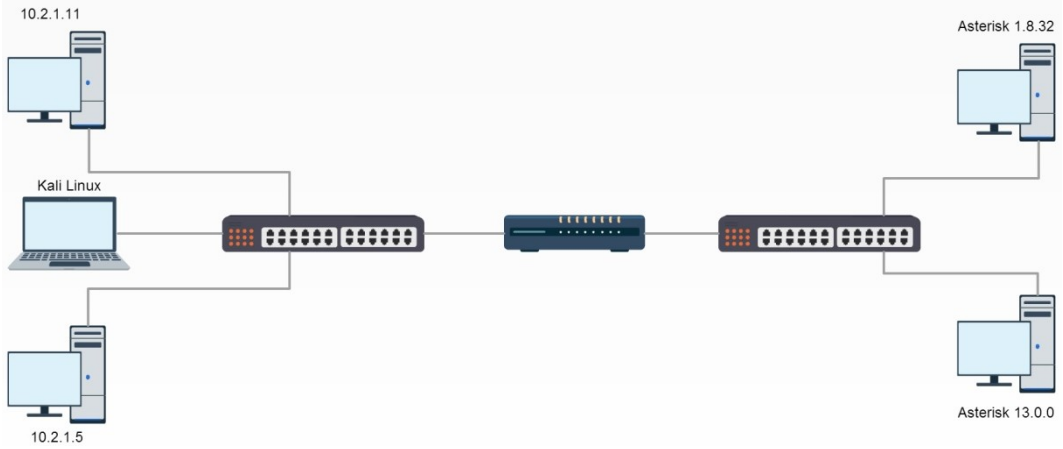
(Applied filter: apply_overrides=1 rows=10 permission=any owner=any sort-reverse=date first=1)

1 - 1 of 1 (total: 1)

Obrázok 4.20: *Výsledok testovania OpenVAS profilom na detekciu hrozby MITM útoku*

4.5.3 **Detekcia hrozby MITM útokov v testovanej topológii s Kali Linux**

Medzi množstvom nástrojov inštalovaných v Kali Linux je aj Ettercap, slúžiaci k útoku typu MITM. Problém je, že tento typ útoku funguje len v prostredí prepínaných sietí, inými slovami v lokálnej sieti. Aby som mohol demonštrovať použitie tohto nástroja, musel som upraviť testovaciu topológiu (obr. 4.21). Notebook som pripojil do siete 10.2.1.0/24, z ktorej je prístup do siete s Asteriskami. Na dvoch počítačoch som spustil softwarového klienta Yate, ktorý podporuje protokol SIP. Na počítači s IP adresou 10.2.1.11 som zaregistroval účet 1001 z Asterisku 1.8.32 a na počítači s IP adresou 13.0.0 som zaregistroval účet 2001 vytvorený na Asterisku 13.0.0.

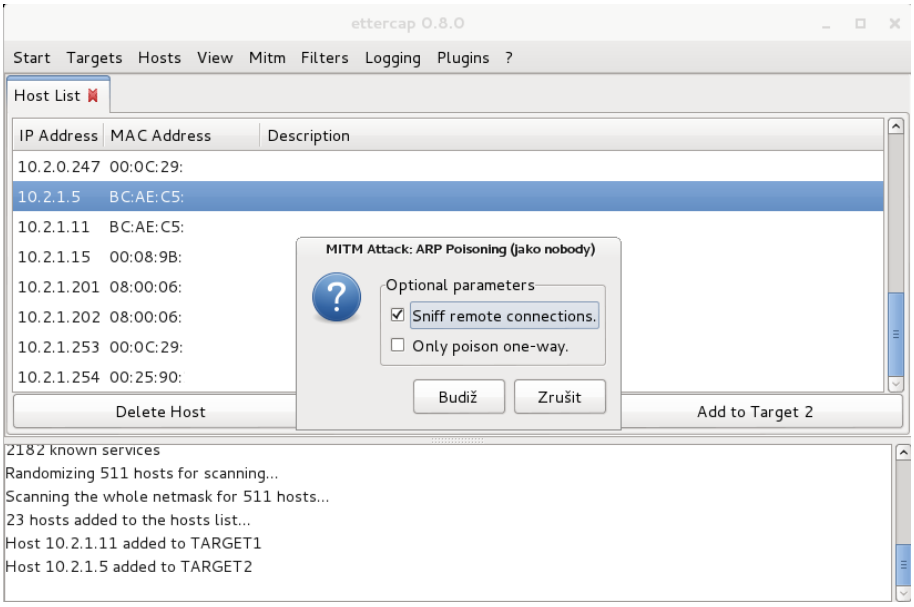


Obrázok 4.21: *Testovacia topológia pri testoch nástrojom MITM z Kali Linux*

V termináli som spustil nástroj Ettercap:

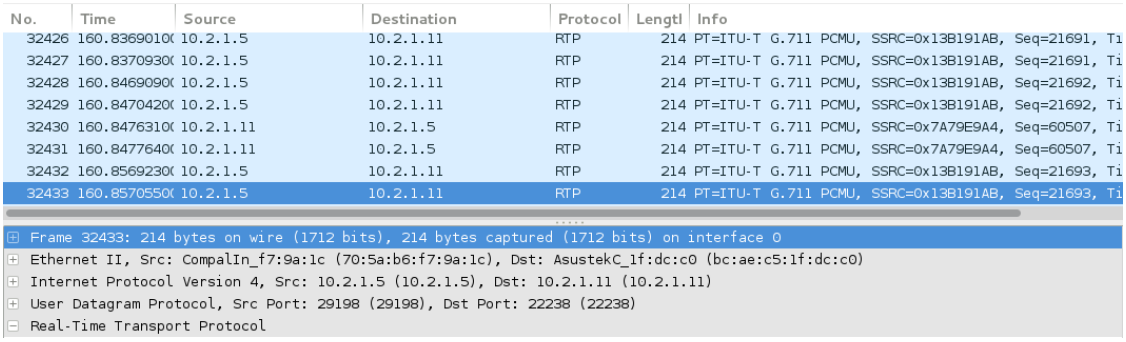
```
ettercap -G
```

Po spustení programu sa zobrazilo úvodné menu. Kliknutím na „Sniff“ a zvolením možnosti „Unified Sniffing“ som sa dostal do voľby sieťového rozhrania. Nasledujúcim krokom bolo vyhľadanie potenciálnych obetí. Súčasným stlačením kláves Crtl a S som spustil skenovanie siete, výsledok skenovania som si zobrazil klávesou H. Zvolil som ciele útoku, čiže počítače s IP adresami 10.2.1.11 a 10.2.1.5 a voľbu potvrdil tlačidlom „Add to Target 1“ resp. „Add to Target 2“. V záložke „Mitm“ som zvolil typ útoku „Arp poisoning...“. Ako je možné vidieť aj na obrázku 4.22, zobrazilo sa malé dialógové okno, v ktorom som povolil „Sniff remote connections“ a potvrdil tlačidlom „OK“. Tým bola konfigurácia dokončená a stačilo už len nástroj spustiť klávesovou skratkou Ctrl a W.



Obrázok 4.22: *Konfigurácia nástroja Ettercap na útok typu MITM*

Z účtu 1001 prihlásenom na počítači s IP adresou 10.2.1.11 som vytvoril hovor na účet 2001 , ktorý bol prihlásený na počítači s IP adresou 10.2.1.5 a pomocou paketového analyzátora Wireshark začal zachytávať prechádzajúcu komunikáciu (obr. 4.23). To bolo možné vďaka tomu, že logika protokolu SIP je uložená v koncových bodoch, takže tie komunikujú priamo medzi sebou a nie cez ústredňu/ústredne.



Obrázok 4.23: *Komunikácia zachytená nástrojom Wireshark počas útoku typu MITM nástrojom Ettercap*

4.5.4 Vyhodnotenie a porovnanie výsledkov testov detegujúcich hrozbu MITM útokov

Cieľom týchto testov bolo nájsť potenciálne riziká a otestovať zraniteľnosť siete na útoky typu MITM.

Nessus testovaním zistil, že v Asterisku verzie 1.8.32 je implementovaná chyba, ktorú by prípadný útočník mohol zneužiť na útok typu MITM. Zvyšné tri odhalené chyby sa týkali starej verzie SSH a podpory nedostatočne bezpečného šifrovania. Tieto tri chyby som do tejto kategórie priradil tiež, pretože SSH sa používa na zabezpečenie spojenia a v prípade, že

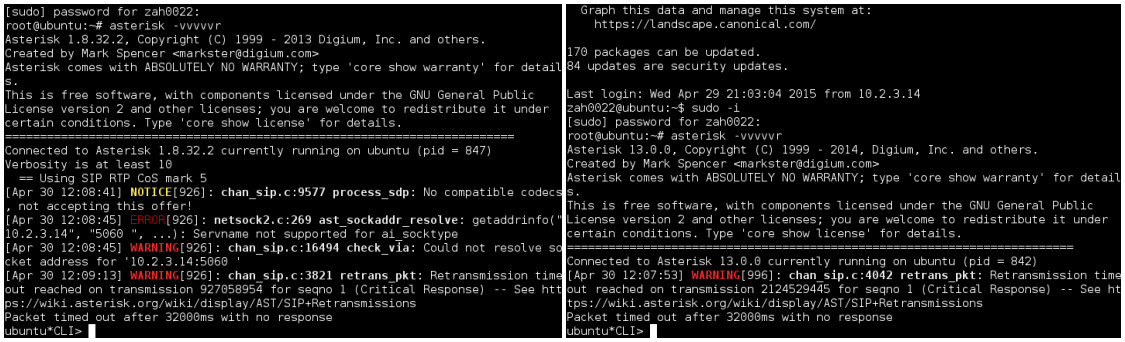
zabezpečenie nie je dostatočné a útočník dokáže správu dešifrovať a prečítať, splní sa tak jeden z cieľov útoku typu MITM, a to čítať komunikáciu medzi dvoma stranami.

OpenVAS s testovanej topológii nevyhľadal žiadne bezpečnostné riziko s možnosťou spôsobenia útoku MITM.

Kali Linux má aj na takéto testy v zálohe pripravený nástroj, no tento nevyhľadáva chyby v kóde, ale priamo testuje zabezpečenie siete proti tomuto druhu útokov. Nakoľko testovacia topológia, a ani jej upravená verzia neobsahujú žiadny bezpečnostný prvok, takýto útok bol samozrejme úspešný.

V testovacej topológii bol teda Nessus schopný detegovať 7 chýb z kategórie útokov typu MITM, OpenVAS nenašiel žiadnu chybu a Kali Linux by simulovaním útoku mohol zaútočiť na ktorýkoľvek z Asteriskov.

Pri spustení Asteriskov v móde na odstraňovanie chýb, boli zobrazované výpisy len v prípade testovania nástrojom OpenVAS (obr. 4.24), činnosť zvyšných dvoch nástrojov pri týchto testoch touto metódou detegovať nebolo možné.



Obrázok 4.24: *Výpis z Asteriskov 1.8.32 (vľavo) a 13.0.0 (vpravo) počas testovania nástrojom OpenVAS s profilom testovania na detegovanie rizika MITM útokov*

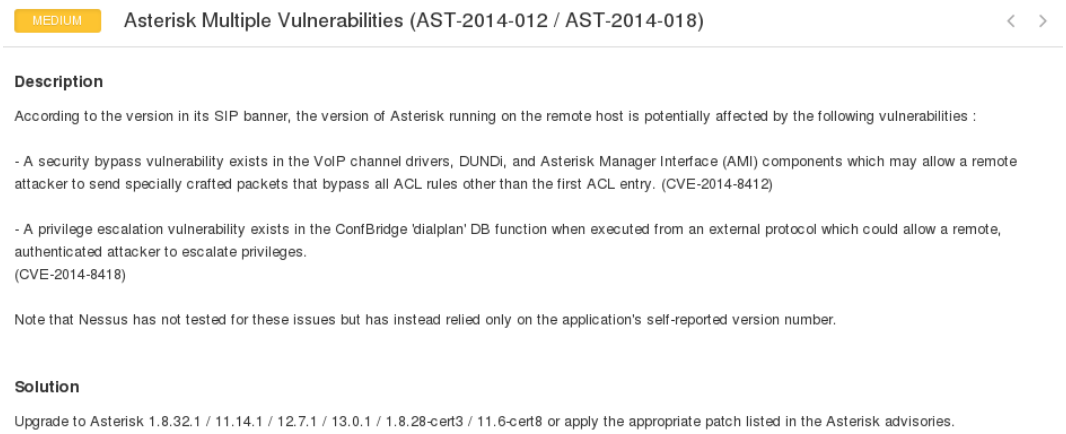
4.6 Penetračné testovanie na detekciu možnosti hrozby SPIT

Spam cez internetovú telefóniu alebo aj hlasový spam je definované ako neželané hromadné zasielanie správ na telefóny s prístupom k internetu.

4.6.1 Detekcia hrozby SPIT útoku v testovanej topológii Nessusom

Nessus neobsahuje žiaden tvorcami vopred pripravený profil a ani žiadnu rodinu pluginov so zameraním na tento typ útoku, takže bolo nutné testovať topológiu profilom vyhľadávajúcim všetky druhy bezpečnostných medzier. Keďže podobná situácia nastala aj v prípade testovania topológie Nessusom na detekciu hrozby MITM útokov, stačilo vziať tento profil a trochu ho upraviť. V sekcii Report som zakázal ukrývať výsledky zo závislých pluginov a spustil som testovanie. Nessus opäť našiel niekoľko bezpečnostných rizík, všetky až na jedno jediné boli spomenuté v predchádzajúcich typoch útokov.

Plugin s ID 73439 objavil chybu v module 'res_pjsip_acl' z Asterisku 13.0.0, ktorá môže útočníkovi pomôcť obísť prístupový zoznam, ktorý by mohol zamietať žiadosti on niektorých užívateľov (obr. 4.25).


MEDIUM Asterisk Multiple Vulnerabilities (AST-2014-012 / AST-2014-018)

Description

According to the version in its SIP banner, the version of Asterisk running on the remote host is potentially affected by the following vulnerabilities :

- A security bypass vulnerability exists in the VoIP channel drivers, DUNDi, and Asterisk Manager Interface (AMI) components which may allow a remote attacker to send specially crafted packets that bypass all ACL rules other than the first ACL entry. (CVE-2014-8412)

- A privilege escalation vulnerability exists in the ConfBridge 'dialplan' DB function when executed from an external protocol which could allow a remote, authenticated attacker to escalate privileges.
(CVE-2014-8418)

Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

Solution

Upgrade to Asterisk 1.8.32.1 / 11.14.1 / 12.7.1 / 13.0.1 / 1.8.28-cert3 / 11.6-cert8 or apply the appropriate patch listed in the Asterisk advisories.

Obrázok 4.25: *Popis chýb v plugine s ID 73439*

4.6.2 Detekcia hrozby SPIT útoku v testovanej topológii OpenVAS

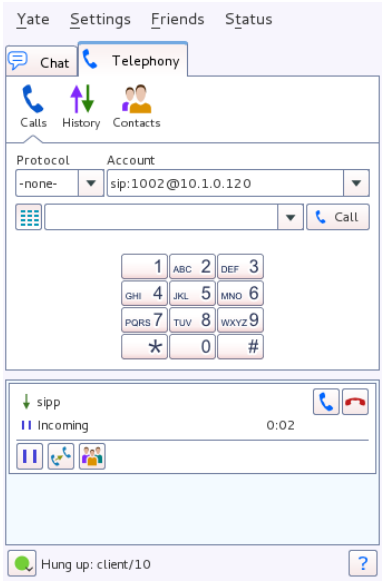
OpenVAS ako mladší a vývojovo „zaostalejší“ derivát Nessusu taktiež nemá žiaden prednastavený profil testovania ani rodinu pluginov zameranú na testovanie sieťových medzier ústiacich v útok typu SPIT. Nakoľko som však už profil testovania, upravený na vyhľadávanie bezpečnostných medzier všetkých typov, konfiguroval aj v prípade detekcie hrozby MITM útokov v testovacej topológii a výsledkom týchto testov bolo, že OpenVAS nenašiel žiadne bezpečnostné riziko, nemalo zmysel opakovať tú istú činnosť a očakávať iný výsledok.

4.6.3 Detekcia hrozby SPIT útoku v testovanej topológii Kali Linux

Kali Linux dokáže otestovať bezpečnostné mechanizmy brániace SPIT útokom v sieti nástrojom SIPp. Ako bolo už niekoľko krát spomenuté, SIPp slúži k auditu VoIP infraštruktúry, ale hlavne je to generátor SIP premávky. Pri spustení nástroja príkazov v tvare:

```
sipp uac -aa -r 1 -rp 300s IP_adresa_SIP_telefónu
```

Začne SIPp vytvárať každých 5 minút na telefón s danou IP adresou nový hovor. Nástroj som takto aj sám otestoval s tým, že som ho nechal vytvárať hovory na účet 1002 z Asterisku verzie 1.8.32 (obr. 4.26). Na tento účet som sa prihlásil cez softwarový telefón Yate.



Obrázok 4.26: *Nástroj SIPp vytvárajúci nové hovory*

4.6.4 Vyhodnotenie a porovnanie výsledkov testov detegujúcich hrozbu SPIT útokov

Cieľom testov bolo overiť bezpečnosť testovanej topológie a schopnosti jednotlivých penetračných nástrojov pri detegovaní hrozieb SPIT útokov.

Výsledkom testov vyhľadávajúcich bezpečnostné medzery, ktoré môžu potenciálne zjednodušiť útočníkovi SPIT útok bola v prípade Nessusu jedna objavená chyba v kóde. OpenVAS nenašiel na testovanej topológii žiadne ohrozenie bezpečnosti týkajúce sa hlasového spamu. Kali Linux poskytol aj v poslednej kategórii útokov iný pohľad na vec, kedy prostredníctvom nástroja SIPp som simuloval správanie útočníka spamujúceho obeť nežiaducimi hovormi.

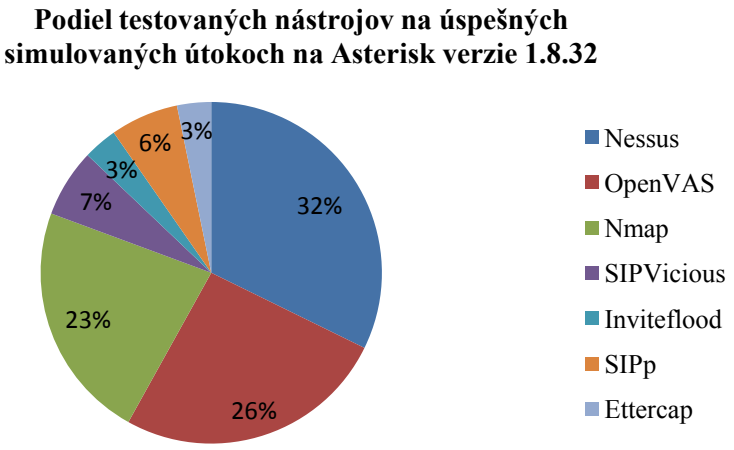
Vyhľadávanie hrozby hlasového spamu Nessusom bolo obom Asteriskom utajené, resp. nehlásili počas testov žiadnu nezvyčajnú činnosť. OpenVAS by sa prejavil tak isto, ako v prípade detegovania ohrozenia útokmi typu MITM. Cieľom SIPp boli telefóny a nie ústredne, takže pomocou výpisov z Asteriskov v debug móde by takýto útok nebolo možné spozorovať.

4.7 Celkové vyhodnotenie výsledkov testov

Hoci ani jedna z rodín pluginov Nessusu nie je venovaná protokolu SIP, v niektorých z nich (napr. Misc.) je možné nájsť hneď niekoľko pluginov testujúcich prvky SIP infraštruktúry. Nebolo by presné povedať, že OpenVAS nedokáže otestovať prvky SIP infraštruktúry, pravdou však je, že obsahuje výrazne menej pluginov, takže v podpore penetračných testov SIP infraštruktúry za Nessusom aj Kali Linux výrazne zaostáva. Kali Linux obsahuje niekoľko nástrojov venovaných výhradne testovaniu prvkov SIP infraštruktúry a niekoľko nástrojov použiteľných nielen pri penetračnom testovaní SIP infraštruktúry. Pri

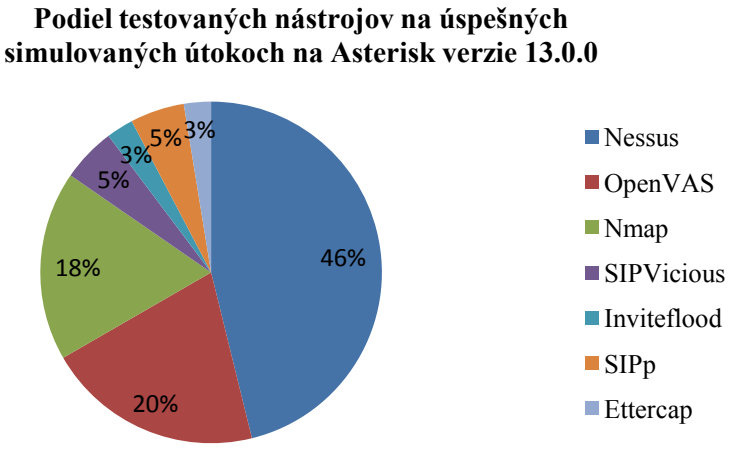
konfigurácii jednotlivých profilov skenovaní som vychádzal z kníh Learning Nessus for Penetration Testing a Kali Linux Cookbook, v ktorej je obsiahnutý aj OpenVAS.

Vzal som všetky výsledky penetračných testov z jednotlivých analyzovaných nástrojov zo všetkých kategórii útokov a graficky ich spracoval tak, aby bolo možné vidieť podiel jednotlivých nástrojov na výslednom množstve odhalených bezpečnostných nedostatkov.



Obrázok 4.27: *Grafické znázornenie podielu testovaných nástrojov na úspešných útokoch na Asterisk verzie 1.8.32*

Totožným spôsobom som porovnal nástroje aj v prípade simulácií útokov na Asterisk verzie 13.0.0:

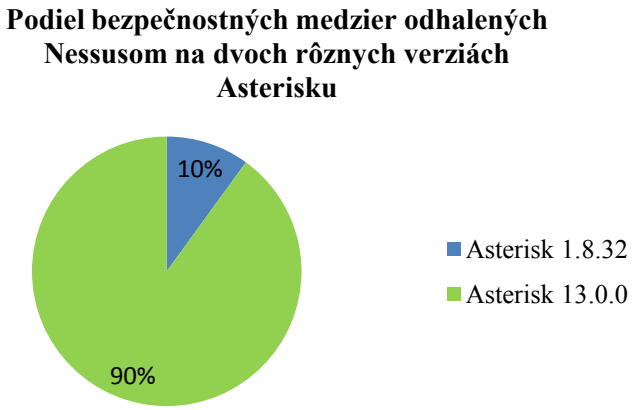


Obrázok 4.28: *Grafické znázornenie podielu testovaných nástrojov na úspešných útokoch na Asterisk verzie 13.0.0*

Na základe výsledkov penetračných testov zo všetkých kategórii útokov usudzujem, že najvhodnejšou metódou pre vytvorenie SIP VoIP penetračných testov je metóda kombinujúca použitie nástrojov z Kali Linux a nástroja Nessus. Kali Linux a Nessus pristupujú

k penetračným testom úplne odlišným spôsobom, čo vo výsledku znamená, že sa veľmi vhodne dopĺňajú.

Jedným z cieľov merania bolo určiť či, a ak áno, tak aké veľké sú rozdiely v bezpečnosti medzi Asteriskom verzie 1.8.32 a verzie 13.0.0. Jediné relevantné výsledky v testoch ponúkol Nessus, pretože OpenVAS nenašiel žiadnu chybu či už v kóde, alebo v konfigurácii na ani jednom z Asteriskov a penetračné testy nástrojov z Kali Linux pracovali nezávisle na verzii Asterisku. Celkovo Nessus odhalil 9 bezpečnostných medzier v prípade Asterisku verzie 13.0.0 a jednu bezpečnostnú medzeru v prípade Asterisku 1.8.32. To znamená, že najstaršia, v súčasnosti ešte stále podporovaná verzia Asterisku s aktualizáciami je bezpečnejšia, ako najnovšia verzia, ktorej aktualizácie chýbajú. (V dobe začatia písania tejto práce bola verzia 13.0.0 tou úplne najnovšou, počas písania bolo vydaných niekoľko aktualizácii. Tie som neinštaloval, pretože by to znehodnotilo výsledky meraní.)



Obrázok 4.29: *Grafické porovnanie podiel bezpečnostných medzier objavených nástrojom Nessus na dvoch rôznych verziách Asterisku*

5 Metódy zabezpečenia siete

5.1 Opatrenia proti stopovaniu

Všetkým príkladom Google hackingu je možné zamedziť jednoduchým pridaním názvu spoločnosti do vyhľadávača alebo prehľadaním webových stránok spoločnosti. Aktívnym vyhľadávaním verejne dostupných prihlasovaní sa na UC zariadenia je možné odstrániť prvotné ciele hackerov. Hoci vo väčšine prípadov nie je žiadny dobrý dôvod, prečo by mal byť telefón alebo pobočková ústredňa dostupná cez internet, je vhodné zmeniť všetky prednastavené heslá pre webové prihlasovanie.

5.2 Opatrenia proti skenovaniu

5.2.1 Zamedzenie ICMP pingu

ICMP premávka môže byť neoceniteľným nástrojom pri meraní a diagnostike stavu sieťových zariadení. Neuvážené povolenie ICMP premávky všetkým sieťovým systémom je rizikom pre sieťovú bezpečnosť, avšak niektoré internetovo založené aplikácie oprávnene vyžadujú možnosť odpovedať na ICMP. Preto firewally a systémy prevencie ohrozenia rozlišujú jednotlivé ICMP žiadosti a odpovede, zatiaľ čo osobné firewally blokovaním ICMP premávky pridávajú ďalšiu úroveň zabezpečenia.

5.2.2 Zamedzenie TCP pingu

Niektoré inteligentné zariadenia sieťovej bezpečnosti, akými sú napr. firewally, systémy prevencie ohrozenia, zariadenia detegujúce sieťové anomálie a smerovače dokážu pomôcť s odhalením a blokovaním TCP pingu. Časť z nich úplne blokujú úvodné ACK a SYN pakety, kým zvyšok sa spúšťa prekročením stanoveného prahu v skenovanej premávke a následným umiestnením zdroja na čiernu listinu.

5.2.3 Zamedzenie SNMP skenovaniu

Najjednoduchším spôsobom ako predísť útokom na zariadenia s povoleným protokolom SNMP je zmeniť komunitné reťazce s prívlastkom verejné a čítanie/zápis z továrenského nastavenia, pretože väčšina súčasných hackovacích a bezpečnostných skenerov vyhľadáva práve prednastavené. Ďalšou možnosťou je obmedziť prístup na SNMP porty (UDP porty 161 a 162) len z autorizovanej administratívnej IP adresy. Poslednou alternatívou je v prípade možnosti použitia SNMP 3. verzie.

5.2.4 Zamedzenie skenovaniu portov

Z pohľadu siete je prvým krokom v prevencii proti skenovaniu internej infraštruktúry aplikácia vhodných pravidiel firewallu. Logickým oddelením siete pomocou VLAN môžeme zabrániť skenovaniu jadra UC infraštruktúry (TFTP, DHCP servery, atď.). Množstvo systémov prevencie ohrozenia a stavových firewallov dokáže odhaliť skenovanie portov a následne

umiestniť problematickú IP adresu na čiernu listinu alebo do karantény. Avšak toto je možné len v prípade TCP skenovania, pretože pri UDP skenovaní nie je zložité adresu podvrhnúť.

Z pohľadu užívateľa je najlepšou obranou vypnutie všetkých služieb, ktoré nie sú nevyhnutné a dobre nastavené prístupové práva vo firewalle.

5.2.5 Zamedzenie snímaniu odtlačku prstu

Nanešťastie neexistuje žiadny jednoduchý spôsob ako zabrániť útočníkovi v určovaní zariadenia alebo operačného systému na základe sieťových odpovedí. Preventívne kroky proti skenovaniu portov cez ICPM, TCP a UDP protokoly môžu túto úlohu sťažiť, prihliadnuc na množstvo rôznych metód detekcie by to pravdepodobne nebolo dostatočne účinné riešenie. Najlepším spôsobom obrany je vypnúť na zariadeniach zbytočné porty a služby.

5.3 Opatrenia proti útokom typu DoS

Obrana proti záplavovým útokom typu DoS a DDoS sa skladá z niekoľkých krokov. Neexistuje nástroj, ktorý by bol schopný zabrániť všetkým druhom DoS útokov. Najlepším riešením je hĺbková ochrana UC zariadení, sieťových komponentov a serverov.

5.3.1 QoS riešenia

Medzi rôznymi riešeniami Quality of Service (kvalita služby) sú v súčasnosti najpoužívanejšie DiffServ (Differentiated Services, rozličné služby). V DiffServ sú pakety označované prioritami obsluhy v závislosti na tom, ku ktorej aplikácii patria. Sieťové zariadenia potom riadia uprednostňovanie a doručovanie takýchto paketov. Inými slovami, RTP pakety majú všeobecne vyššiu prioritu ako napr. emailové pakety.

Prioritu paketov je možné označiť niekoľkými spôsobmi. Na IP vrstve sa ponúka využiť pole DSCP (differentiated services code point). Rovnako efektívne, ale oveľa používanejšie sú IEEE štandardy 802.1P a 802.1Q. 802.1P definuje schémy pre uprednostňovanie paketov a hlavička 802.1Q obsahuje pole 802.1P. Avšak tento prístup je možný len v prípade použitia VLAN.

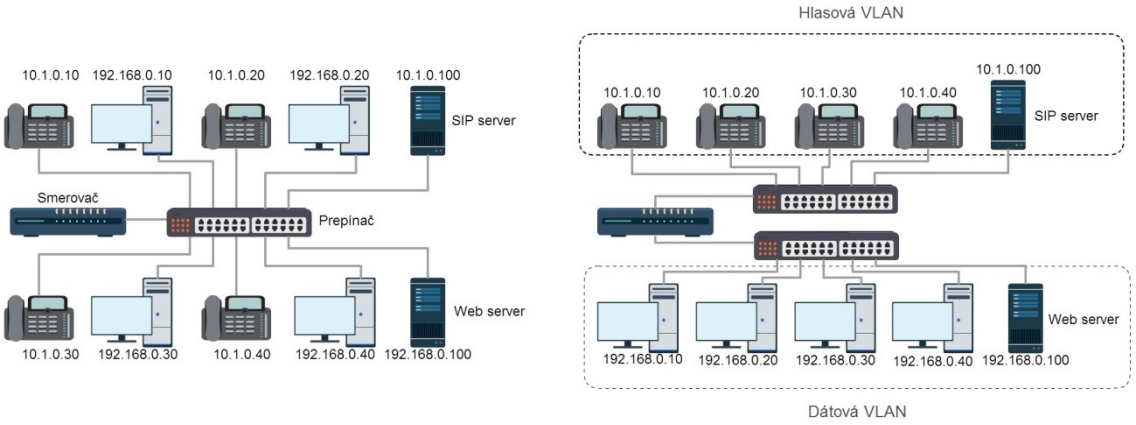
5.3.2 Zvýšenie úrovne zabezpečenia UC telefónov a serverov

Zvýšenie úrovne zabezpečenia UC telefónov a serverov alebo aj tzv. hardening spočíva v základných odporúčaniach nezávisle na konkrétnom výrobcovi:

- zmeniť prednastavené heslá a odstrániť všetky hosťovské a neautentizované účty,
- zastaviť služby, ktoré nie sú nevyhnutné pre chod zariadenia (telnet, http, atď),
- uistiť sa, že zariadenie alebo operačný systém je aktualizovaný s najnovšími záplatami,
- vypracovať stratégiu, ktorá udrží zariadenia aktualizované.

5.3.3 VLANy

Virtuálne lokálne siete sa používajú k logickému oddeleniu sietí na jednom prepínači, čím môžu poskytnúť ochranu pre UC servery a zariadenia (obr. 5.1). Množstvo prepínačov umožňuje vytvoriť súčasne niekoľko virtuálnych lokálnych sietí a pomôcť tak ochrániť UC servery a zariadenia pred väčšinou z DoS premávky, napr. červami a vírusmi. Problém nastáva v prípade používania softwarových telefónov, pretože je ťažké logicky oddeliť UC aplikácie od dátovej siete nakoľko počítač potrebuje prístup k zdrojom dátovej siete. V takom prípade je nutné mať správne nakonfigurované QoS, ktoré v každom prípade uprednostní hlasové pakety pred zvyškom dátovej premávky.



Obrázok 5.1: *Rozdiel medzi fyzickou (vľavo) a logickou (vpravo) topológiou pri použití virtuálnych lokálnych sietí*

5.3.4 NIPS

Network intrustion prevention systems alebo aj systémy prevencie sieťového ohrozenia sú vnútorné sieťové zariadenia, ktoré rýchlosťou linky detegujú a blokujú útoky. NIPS sú do siete nasadzované podobným spôsobom ako prepínače alebo smerovače. S cieľom odhaliť podozrivú aktivitu, prehľadávajú všetky pakety, ktoré cez ne prechádzajú. NIPS sú súčasťou sieťovej infraštruktúry, takže musia rozlišovať medzi útokmi a legitímnou premávkou.

5.3.5 Zamedzenie vyčerpaniu DHCP

Existuje niekoľko spôsobov ako zabrániť útoku vyčerpanie DHCP. DHCP servery je možné nakonfigurovať tak, aby neprenajímali IP adresy pre neznáme MAC adresy alebo pre nedôveryhodné segmenty siete. Cisco prepínače majú funkciu zvanú DHCP snooping, ktorá sa správa ako DHCP firewall rozlišujúci dôveryhodné a nedôveryhodné sieťové rozhrania.

5.4 Opatrenia proti krádeži registrácie

Je možné použiť niekoľko protiopatrení brániacich útočníkovi v manipulácii s registráciou. Ich cieľom je zabezpečiť registračný proces a zabrániť SIP proxy v prijímaní neplatných registrácii.

5.4.1 Použitie TCP protokolu pre SIP spojenia

V súlade s RFC 3261 musia SIP proxy a SIP koncové body podporovať TCP aj UDP protokol. V prípade použitia protokolu TCP zvyčajne koncové body SIP nadväzujú trvalé spojenie. Prihliadnuc na súčasti TCP protokolu akými sú napr. sekvenčné čísla, by tak bolo pre útočníka oveľa zložitejšie prinútiť SIP proxy prijať podvrhnutú registráciu. Aby bola táto metóda účinná, musia všetky koncové body SIP komunikujúce so SIP proxy používať protokol TCP.

V prípade používania TCP protokolu sa navyše ponúka možnosť použiť Transport Layer Security. TLS bráni odpočúvaniu signalizácie šifrovaním dát a poskytuje silnú autentizáciu čím výrazne komplikuje oklamanie SIP proxy podvrhnutou registráciou.

TLS slúži k zabezpečeniu jednotlivých spojení medzi SIP proxy a SIP koncovými bodmi. Avšak TLS nie je koncovým protokolom. Aby bol hovor zabezpečený, musí byť TLS použité vo všetkých spojoch medzi koncovými bodmi zúčastnených v hovore. V prípade, že v čo i len jedinom spoji nie je použité TLS, celý bezpečnostný model stráca zmysel.

5.4.2 Povolenie autentizácie

Zo všetkých SIP žiadostí dáva najväčší zmysel podpora autentizácie pre žiadosť REGISTER. Žiadosti REGISTER nie sú vymieňané často, takže réžia autentizácie je minimálna. Len interné a firemné koncové body SIP by sa mali registrovať, takže je možné povoliť autentizáciu a nastaviť silné heslá pre každý koncový bod. Slabé alebo mechanicky vygenerované heslo by mohol útočník jednoducho uhádnuť alebo prelomiť.

5.4.3 Zníženie registračného intervalu

Znížením registračného intervalu donútime koncové body registrovať sa častejšie. Ak nastavíme registračný interval na 60 sekúnd, tak ak by bola registrácia odstránená alebo ukradnutá, koncový bod by sa do minúty vrátil späť do normálnej prevádzky.

5.4.4 Použitie SBC a SIP firewallov

Hraničný kontrolór relácií (SBC) a SIP firewall môžu byť nasadené na prehľadávanie všetkej signalizácie odosielanej na SIP proxy. SIP firewall deteguje rôzne formy útokov, vrátane útokov manipulujúcich s registráciou. SIP firewally a SBC sú zásadné pri pripájaní do verejnej siete.

5.5 Opatrenia proti útokom typu MITM

5.5.1 Zabezpečenie portov prepínača

Otráveniu ARP možno zabrániť zavedením prísnych bezpečnostných pravidiel na prepínači. Manuálnym vložením zoznamu zdrojových MAC adries povolených pristupovať na jednotlivé porty prepínača je možné znemožniť nezvaným sieťovým uzlom prístup do siete.

Je nutné poznamenať, že zabezpečenie portov nemusí byť dostatočným riešením. V prípade, že by útočník odpojil telefón a na jeho miesto pripojil napr. notebook s podvrhnutou MAC adresou, získal by prístup do siete. Ďalšou nevýhodou je, že zabezpečenie portov obmedzí možnosť presúvať zariadenia naprieč sieťou.

5.5.2 VLANy

VLANy môžu poskytnú ďalšiu úroveň ochrany pred triviálnymi ARP podvrhujúcimi technikami logickým oddelením kritickej UC infraštruktúry od dátovej siete. V niektorých prípadoch útočníkovi navyše ešte znemožnené skenovať legitímne MAC adresy v sieti.

5.5.3 Šifrovanie relácie

Šifrovať a brániť sa tak útoku MITM je možné na niekoľkých vrstvách, napr. použitím IPSec na sieťovej vrstve alebo SRTP a ZRTP na aplikačnej vrstve. Ďalšou možnou alternatívou je povolenie TLS.

5.5.4 Nástroje detegujúce otravu ARP

Takéto nástroje si uchovávajú záznamy z mapovania MAC adries/IP adries a prípadné zmeny hlásia prostredníctvom mailu alebo syslogu.

5.6 Opatrenia proti hlasovému SPAMu

5.6.1 Overenie identity

Jedným z hlavných bodov pri riešení hlasového spamu je schopnosť určiť totožnosť volajúceho. Tá je uvedená v SIP hlavičke v poli „Od:“. Nanešťastie táto položka je jednoducho sfalšovateľná.

Z dôvodu stanovenia identity sa musia všetci užívatelia v SIP doméne autentizovať. RFC 3261 vyžaduje podporu prehľadnej autentizácie (digest authentication). V kombinácii s TLS medzi všetkými SIP koncovými bodmi a SIP proxy môže byť prehľadná autentizácia použitá k bezpečnému overovaniu užívateľských agentov. V prípade, že takýto užívateľský agent odošle hovor do ďalšej domény, preukáže sa jeho identita. Podobne ako pri šifrovaní pomocou TLS, aj pri overovaní identity stráca model zmysel v prípade, že ľubovoľná SIP proxy nepodporuje TLS alebo nie je dôveryhodná.

5.6.2 Podnikové spam filtre

Niektoré spoločnosti riešia problém s hlasovým spamom obdobne ako v prípade emailového spamu, zadovážia si komerčný produkt. Komerčné produkty väčšinou pracujú na základe:

Čierne zoznamy sú zbierkou adries známych útočníkov. Dobre nadefinované a preverené čierne zoznamy môžu byť použité pri hovorov od známych hlasových spamerov.

Biele zoznamy sú zbierkou adries od ktorých chce užívateľ prijímať volania. Pri používaní bielych zoznam si užívateľ sám pridáva nové zdroje od ktorých chce prijímať hovory. Nie je spôsob akým by sa útočník mohol dostať na biely zoznam, ak však pozná nejakú adresu zo zoznamu, môže svoju sfalšovať a vytvárať hovory. [14]

Schvaľovací systém spolupracuje s bielym aj čiernym zoznamom. V prípade, že sa nový volajúci snaží vytvoriť hovor, užívateľ je vyzvaný k reakcii. Prijatím hovoru volajúceho umiestni na biely zoznam, zamietnutím hovoru na čierny zoznam. Nevýhodou je možnosť zaplavenia užívateľa žiadosťami o schválenie.

Filtrovanie zvukového obsahu môže byť nasadené až v prípade, že bol nejaký hlasový spam uložený do hlasovej schránky. Po prevode reči na text je možné obsah analyzovať a vyhľadávať tak hlasový spam. Tieto správy potom zvyčajne končia v koši.

Hlasový CAPTCHA/Turingov test sú úlohy, na ktoré dokáže odpovedať jednoducho iba človek. Príkladom môže byť textová správa ukrytá v obrázku, ktorú väčšina ľudí jednoducho prečíta, ale stroj to vykonať nedokáže. Hlasový CAPTCHA pracuje podobne. V prípade prichádzajúceho hovoru je volaný uvítaný nejakou úlohou ako napr.: „Prosím zadajte prvé tri písmená z mena volaného“. Takéto testy sú veľmi jednoduché pre ľudí, ale náročné pre počítač. V prípade, že volaný odpovie správne, je hovor odoslaný užívateľovi. Ak úlohu nesplní, hovor je zamietnutý alebo presmerovaný do hlasovej schránky, prípadne do koša. Negatívnou stránkou je obťažovanie legitímnych volajúcich. Obzvlášť v prípade, že volajúci musí výzvu z nejakého dôvodu podstúpiť opakovane niekoľko krát za sebou. Najvýhodnejšie je nasadiť CAPTCHA v spolupráci s bielym a čiernym zoznamom, kedy úlohy podstupujú len nový alebo podozrivý volajúci. [18]

5.7 Príklady možností zabezpečenia testovanej siete

Pri praktických príkladoch možností ako zabezpečiť testovanú sieť som si zvolil zariadenia od spoločnosti Cisco, pretože boli odporúčané v knihe HACKING EXPOSED: Unified Communications & VoIP Security Secrets & Solutions a aj z dôvodu osobnej skúsenosti s konfiguráciou týchto zariadení.

5.7.1 Zamedzenie ICMP pingu a SNMP skenovaniu

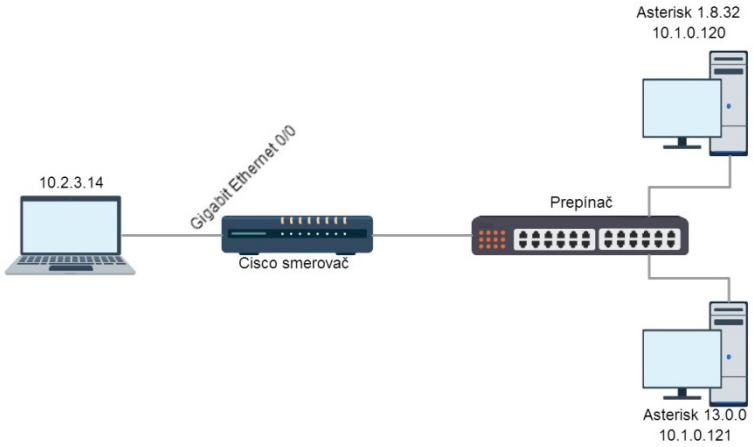
Zabrániť zariadeniam z internetu, resp. z vonkajších sietí v odosielaní ICMP žiadostí a v SNMP skenovaní je možné jednoduchou konfiguráciou prístupového zoznamu (access-list). V prípade testovanej topológie by bolo najvhodnejšie umiestniť takýto prístupový zoznam na rozhranie smerovača vedúceho k môjmu počítaču (obr. 5.2). Konfigurácia takéhoto prístupového zoznamu:

```
Smerovac(config)#access-list 101 deny icmp any any
Smerovac(config)#access-list 101 deny udp any any eq snmp
Smerovac(config)#access-list 101 permit ip any any
```

Na záver je nutné prístupový zoznam priradiť k sieťovému rozhraniu:

```
Smerovac(config)#int GigabitEthernet 0/0
Smerovac(config-if)#ip access-group 101 in
```

Po príchode akejkoľvek premávky na rozhranie GigabitEthernet 0/0 ju smerovač skontroluje prístupovým zoznamom s číslom 101. V prípade príchodu akejkoľvek ICMP alebo SNMP premávky je takáto premávka zahodená, všetka ostatná premávka je smerovaná ďalej do vnútornej siete.



Obrázok 5.2: *Testovaná topológia s Cisco smerovačom*

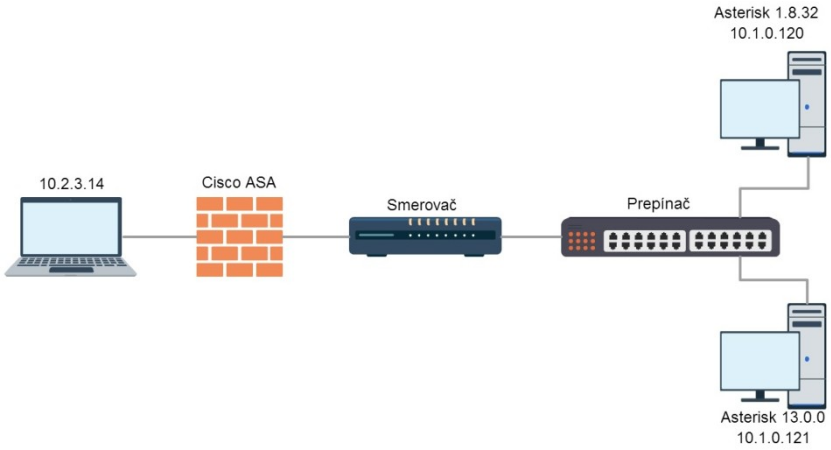
5.7.2 Zamedzenie TCP SYN skenovaniu a TCP SYN záplavovému útoku

K zamedzeniu TCP SYN útokov je nutné použiť pokročilejší prístup než ponúka prístupový zoznam. Jednou z možností je nasadiť Cisco ASA (obr. 5.3). Cisco ASA je stavový

firewall a VPN koncentrátor v jednom. Ukážková konfigurácia Cisco ASA tak, aby jednej IP adrese bolo umožnené vytvoriť maximálne päť TCP SYN spojení:

```
CiscoASA(config)#access-list 101 permit tcp any 10.1.0.0
255.255.255.0
CiscoASA(config)#class-map Premavka_do_siete
CiscoASA(config-cmap)#match access-list 101
CiscoASA(config-cmap)#exit
CiscoASA(config)#policy-map Vseobecne_pravidla
CiscoASA(config-pmap)#class Premavka_do_siete
CiscoASA(config-pmap-c)#set connection embryonic-conn-max 5
```

Prvým príkazom je vytvorený prístupový zoznam identifikujúci premávku do siete 10.1.0.0/24. Nasleduje vytvorenie triedy Premavka_do_siete mapujúcej prístupový zoznam 101. Poslednými troma príkazmi sa vytvorí mapa pravidiel Vseobecne_pravidla, ktorá povolí triede Premavka_do_siete vytvoriť maximálne päť nedokončených TCP spojení. [8]



Obrázok 5.3: *Nasadenie Cisco ASA stavového firewallu do testovanej siete*

5.7.3 Zabránenie prístupu do siete zabezpečením portov prepínača

(Nielen) Cisco smerovače dokážu zabrániť útočníkovi, ktorý získal fyzický prístup do siete zabezpečením portov prepínača:

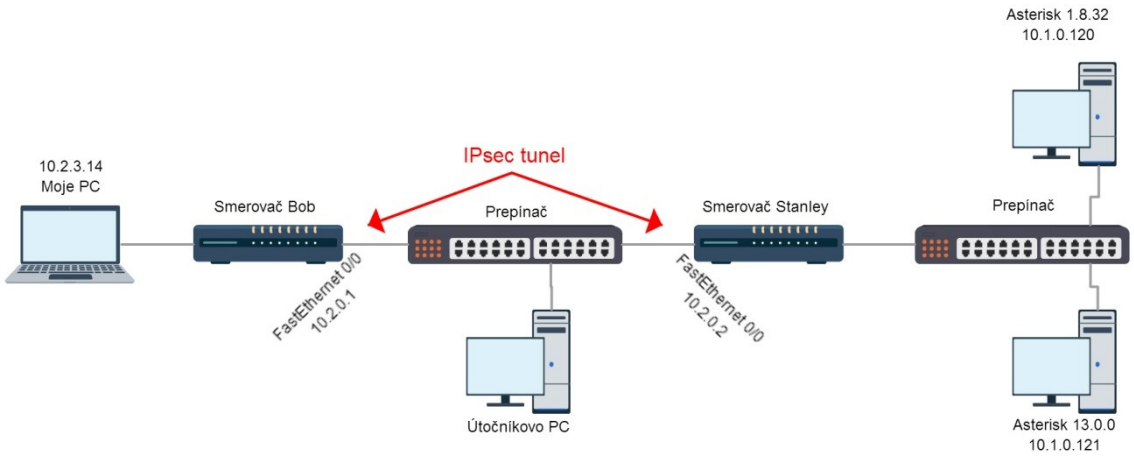
```
Prepinac(config)#interface range FastEthernet 0/10-20
Prepinac(config-if-range)#switchport mode access
Prepinac(config-if-range)#switchport port-security
Prepinac(config-if-range)#switchport port-security violation
shutdown
Prepinac(config-if-range)#exit
```

```
Prepinac(config)#interface FastEthernet 0/10
Prepinac(config-if)#switchport port-security mac-address
aabb.ccdd.eeff
```

Príklad konfigurácie zabezpečenia portov prepínača začína zvolením rozsahu zabezpečovaných rozhraní. Tieto rozhrania sú v móde access, čo znamená, že vedú ku koncovým zariadeniam. Ďalej konfigurácia obsahuje príkaz, ktorý prikazuje v prípade pripojenia nepovoleného zariadenia port odpojiť. Na záver ukážka priradenia MAC adresy AA:BB:CC:DD:EE:FF na rozhranie FastEthernet 0/10.

5.7.4 IPsec tunel ako obrana proti MITM útoku

Ďalším možným, aj keď nepravdepodobným scenárom v testovanej topológii kedy by útočník mohol sledovať dáta odosielané medzi mojim počítačom a virtuálnymi počítačmi s Asteriskami je útok MITM, konkrétne Otrava ARP (obr. 5.4). (Tento scenár je nepravdepodobný pretože smerovače, predovšetkým tie drahšie majú implementované nástroje ako protiopatrenie tomuto typu útoku.)



Obrázok 5.4: *MITM útok v testovanej topológii*

Dáta prechádzajúce cez sieť uprostred je možné zabezpečiť pomocou technológie IPsec. Bezpečnostné služby poskytujú dva protokoly:

- Encapsulating Security Payload (ESP),
- Authentication Header (AH).

ESP poskytuje paketom dôvernosť, autentizáciu, integritu prenášaných dát a ochranu proti útokom využívajúcim opätovné posielanie paketov. AH poskytuje paketom autentizáciu, integritu prenášaných dát a ochranu proti útokom využívajúcim opätovné posielanie paketov. Autentizácia sa vzťahuje aj časť IP záhlavia, ale AH dáta nešifruje. V praxi sa bežne používa kombinácia týchto protokolov.

Na úvod je nutné definovať spôsob autentizácie a šifrovania prenášaných dát:

```
SmerovacBoB(config)#crypto isakmp policy 10
SmerovacBoB(config-isakmp)#authentication pre-share
SmerovacBoB(config-isakmp)#encryption aes 256
SmerovacBoB(config-isakmp)#group 5
SmerovacBoB(config-isakmp)#hash sha
SmerovacBoB(config-isakmp)#lifetime 3600

SmerovacStanley(config)#crypto isakmp policy 10
SmerovacStanley(config-isakmp)#authentication pre-share
SmerovacStanley(config-isakmp)#encryption aes 256
SmerovacStanley(config-isakmp)#group 5
SmerovacStanley(config-isakmp)#hash sha
SmerovacStanley(config-isakmp)#lifetime 3600
```

Ďalším krokom je definícia hesla a druhého konca tunelu:

```
SmerovacBob(config)#crypto isakmp key HESLO123 address 10.2.0.2
SmerovacStanley(config)#crypto isakmp key HESLO123 address
10.2.0.1
```

Nasleduje stanovenie paketov, ktoré budú prechádzať tunelom a budú šifrované:

```
SmerovacBob(config)#access-list 101 permit ip host 10.2.3.14
10.1.0.0 0.0.0.255
SmerovacStanley(config)#access-list 101 permit ip 10.1.0.0
0.0.0.255 host 10.2.3.14
```

Vytvorenie kryptovacej mapy s názvom DiplomovaPraca:

```
SmerovacBob(config)#crypto map DiplomovaPraca 5 ipsec-isakmp
SmerovacBob(config-crypto-map)#match address 101
SmerovacBob(config-crypto-map)#set peer 10.2.0.2
SmerovacBob(config-crypto-map)#set transform-set 50

SmerovacStanley(config)#crypto map DiplomovaPraca 5 ipsec-isakmp
SmerovacStanley(config-crypto-map)#match address 101
SmerovacStanley(config-crypto-map)#set peer 10.2.0.1
```

```
SmerovacStanley(config-crypto-map)#set transform-set 50
```

Posledným krokom je priradenie kryptovacej mapy na rozhrania smerovačov:

```
SmerovacBob(config)#interface fastethernet 0/0
SmerovacBob(config-if)#crypto map DiplomovaPraca

SmerovacStanley(config)#interface fastethernet 0/0
SmerovacStanley(config-if)#crypto map DiplomovaPraca
```

Záver

Cieľom diplomovej práce bolo analyzovať súčasný stav nástrojov realizujúcich penetračné testovanie, predovšetkým z pohľadu podpory penetračných testov pre SIP prvky IP telefónie, realizovať zvolenými nástrojmi penetračné testy a na základe výsledkov testov stanoviť najvhodnejšiu metódu testovania a zásady zabezpečenia. V úvode práce som čitateľa oboznámil so základnými pojmami z oblasti internetových protokolov, bežnými spôsobmi zabezpečovania prenosu sprostredkúvaného týmito protokolmi. Ďalšia kapitola obsahuje popis jednotlivých typov útokov, vyskytujúcich sa v IP telefónii. Cieľom tretej kapitoly bolo poskytnúť čitateľovi základný prehľad o aplikáciách používaných pri vytváraní tejto práce. Účelom prvých troch kapitol tejto diplomovej práce bolo priblížiť čitateľovi problematiku penetračných testov s IP telefónii. V štvrtej kapitole boli realizované penetračné testy prostredníctvom všetkých troch zvolených nástrojov rozdelené podľa kategórii útokov popísaných v druhej kapitole. V tejto kapitole však bolo možné nájsť aj konfigurácie profilov skenovaní, porovnanie jednotlivých nástrojov stanovené na základe výsledkov testov a taktiež porovnanie dvoch rôznych verzií Asterisku. V poslednej kapitole som zúročil všetky vedomosti nadobudnuté štúdiom a tvorbou predchádzajúcich kapitol pri vytváraní zásad zabezpečenia siete pred útokmi vykonávanými analyzovanými nástrojmi. Aplikáciou týchto pravidiel v praxi by sa VoIP infraštruktúra mala stať zabezpečenou.

Prihliadnuc na výsledky obsiahnuté v tejto práci, na množstvo rôznych stále pribúdajúcich variant útokov a flexibilitu útočníkov, považujem za nevyhnutné v podobnom výskume pokračovať a rozšíriť testy o ďalšie prvky SIP infraštruktúry, predovšetkým o tie, na ktorých útok by mohol mať kritické dôsledky.

Použitá literatúra

[1] PRITCHETT, Willie L a David DE SMET. *Kali Linux cookbook*. Birmingham: Packt Publishing, 2013. ISBN 978-1-78328-960-8.

[2] ENDLER, David a Mark COLLIER. *HACKING EXPOSED: Unified Communications & VoIP Security Secrets & Solutions. 2nd edition*. New York: McGraw-Hill Education, 2014. ISBN 978-007-1798-778.

[3] ENDLER, David a Mark COLLIER. *Hacking Exposed VoIP: Voice Over IP Security Secrets & Solutions.* McGraw-Hill, 2007. ISBN 9780072263640.

[4] LYON, Gordon. *Nmap Network Scanning: Official Nmap Project Guide to Network Discovery and Security Scanning.* Sunnyvale, California: Insecure.Com LLC, 2008. ISBN 978-0-9799587-1-7.

[5] GARFINKEL, Simson a Gene SPAFFORD. *Practical UNIX & Internet Security.* USA: O´Reilly, 1996. Second Edition. ISBN 15-659-2148-8.

[6] CHAPPELL, Laura. *Wireshark 101: Essential Skills for Network Analysis.* First Edition. San Jose, California. ISBN 978-1-893939-73-8.

[7] KUMAR, Himanshu. *Learning Nessus for penetration testing.* Birmingham, England: Packt Publishing, 2014. ISBN 978-1-78355-100-2.

[8] CISCO SYSTEMS, Inc. Cisco *ASA Series CLI Configuration Guide* [online]. 2012 [cit. 2015-05-01]. Dostupné z: cisco.com

[9] NORTON, Duane. SANS INSTITUTE. An *Ettercap Primer* [online]. 2004 [cit. 2015-05-01]. Dostupné z: sans.org

[10] WAGNER, Jan - Oliver, Michael WIEGAND, Tim BROWN a Carsten Koch MAUTHE. *OpenVAS Compendium* [online]. Intevation GmbH, 2008 [cit. 2015-05-01]. Dostupné z: openvas.org

[11] TENABLE. *Tenable Products Plugin Families* [online]. Columbia, USA, 2013 [cit. 2015-05-01]. Dostupné z: tenable.com

[12] VOZŇÁK, Miroslav a Filip ŘEZÁČ. VŠB-TU OSTRAVA. *Asterisk teorie a praxe* [online]. Ostrava, Česká Republika, 2011 [cit. 2015-05-01].

[13] VOZŇÁK, Miroslav. *Voice over Internet Protocol* [online]. Ostrava, 2014 [cit. 2015-05-01].

[14] VOZŇÁK, Miroslav a Filip ŘEZÁČ. VŠB-TU OSTRAVA. *Security Risks in IP Telephony* [online]. First Edition. Ostrava, 2010 [cit. 2015-05-01].

[15] VOZŇÁK, Miroslav. *Signalizace SIP* [online]. Praha, 2006 [cit. 2015-05-01]. Dostupné z: http://www.phonet.cz/archiv/dok_osta/ipt-2006_Signalizace_SIP.pdf

[16] ANDERSON, Harry. SYMANTEC. *Introduction to Nessus* [online]. 2010 [cit. 2015-05-01]. Dostupné z:http://www.symantec.com/connect/articles/introduction-nessus

[17] CERT. *Denial of Service Attacks* [online]. 1997 [cit. 2015-05-01]. Dostupné z: http://www.cert.org/information-for/denial_of_service.cfm?

[18] KHAYARI, Rachid El. TECHNISHE UNIVERSITÄT DARMSTADT. *SPAM over Internet Telephony and how to deal with it* [online]. [cit. 2015-05-01].

[19] HANDLEY a JACOBSON. IETF. *SDP: Session Description Protocol* [online]. 1998 [cit. 2015-05-01]. Dostupné z: https://www.ietf.org/rfc/rfc2327.txt

[20] UNIVERZITY OF MASSACHUSETTS. *Connectionless Transport: UDP* [online]. 2000 [cit. 2015-05-01]. Dostupné z: http://www-net.cs.umass.edu/kurose/transport/UDP.html

[21] KEFER, Daniel. *IW: Penetračné testy – úvod do legálneho hackingu* [online]. 2010 [cit. 2015-05-01]. Dostupné z:http://www.itnews.sk/tituly/infoware/2010-04-30/c133372-iw-penetracne-testy-uvod-do-legalneho-hackingu

Zoznam príloh

Príloha A: *Súbor sip.conf z Asterisku verzie 1.8.32*

```
[asterisk0]
type=friend
host=10.1.0.121
context=incoming
insecure=invite
disallow=all
allow=ulaw

[1001]
type=friend
secret=1001
userid=1001 <1001>
host=dynamic
context=internal

[1002]
type=friend
secret=1002
userid=1002 <1002>
host=dynamic
context=internal
```

Príloha B: *Súbor extensions.conf z Asterisku verzie 1.8.32*

```
[internal]
exten => 1001,1,Dial(SIP/1001,20)
exten => 1001,2,Hangup
exten => 1002,1,Dial(SIP/1002,20)
exten => 1002,2,Hangup
exten => _2XXX,1,Dial(SIP/${EXTEN}@asterisk0,20)
exten => _2XXX,2,Hangup

[incoming]
exten => 1001,1,Dial(SIP/1001,20)
exten => 1002,1,Dial(SIP/1002,20)
```

Príloha C: *Súbor sip.conf z Asterisku verzie 13.0.0*

```
[asterisk1]
type=friend
host=10.1.0.120
context=incoming
insecure=invite
disallow=all
allow=ulaw

[2001]
type=friend
secret=2001
userid=2001 <2001>
host=dynamic
context=internal

[2002]
type=friend
secret=2002
userid=2002 <2002>
host=dynamic
context=internal
```

Príloha D: *Súbor extensions.conf z Asterisku verzie 13.0.0*

```
[internal]
exten => 2001,1,Dial(SIP/2001,20)
exten => 2001,2,Hangup
exten => 2002,1,Dial(SIP/2002,20)
exten => 2002,2,Hangup
exten => _1XXX,1,Dial(SIP/${EXTEN}@asterisk1,20)
exten => _1XXX,2,Hangup

[incoming]
exten => 2001,1,Dial(SIP/2001,20)
exten => 2002,1,Dial(SIP/2002,20)
```